

MỤC LỤC

MỤC LỤC	1
1. HƯỚNG DẪN CẤU HÌNH AUTO BUILD PROJECT DẠNG ANT	5
Bước 1: Tạo một job tương ứng cho dự án.	5
Bước 2: Cấu hình chung cho job	5
Bước 3: Cấu hình đường dẫn tới bản source code	8
Bước 4: Cấu hình build job	13
Bước 5: Cấu hình đẩy artifact tới Nexus Server	15
Bước 6: Cấu hình Auto Deploy	17
Bước 7: Cấu hình gửi email thông báo khi build faild	19
2. HƯỚNG DẪN CẤU HÌNH AUTO BUILD ĐỐI VỚI PROJECT DẠNG MAVEN	21
Bước 1: Tạo mới một project tương ứng với dự án.	21
Bước 2: Cấu hình Job	21
Bước 3: Cấu hình push artifact tới nexus server	26
Bước 4: Chạy Job và check kết quả	27
Bước 5: Cấu hình auto deploy sau khi build	28
Bước 6: Cấu hình gửi email thông báo khi build faild	29
3. HƯỚNG DẪN CẤU HÌNH AUTO BUILD PROJECT DẠNG MSBUILD	31
Bước 1: Tạo một project tương ứng với dự án	31
Bước 2: Cấu hình job	32
Bước 3: Cấu hình push artifact tới nexus server	36
Bước 4: Chạy job và check kết quả	37
Bước 5: Cấu hình gửi email thông báo khi build faild	38
Bước 6: Cấu hình tham số sử dụng cho job thực hiện Auto Deploy:	40
Bước 7: Cấu hình job thực hiện Auto Deploy	42
4. HƯỚNG DẪN CẤU HÌNH BUILD VỚI PROJECT ANGULAR	44
Bước 1: Tạo một project tương ứng với dự án	44
Bước 2: Cấu hình job	45
Bước 3: Cấu hình push artifact tới Nexus server	47
Bước 4: Chạy job và check kết quả	47
Bước 5: Cấu hình gửi email thông báo	48
Bước 6: Cấu hình job thực hiện Auto Deploy	50

5.	Hướng dẫn cấu hình build với project Android sử dụng gradle build	50
5.1.	<i>Tạo một project tương ứng với dự án</i>	50
5.2.	<i>Cấu hình job</i>	50
6.	HƯỚNG DẪN VIẾT ANSIBLE PLAYBOOK ĐỂ AUTO DEPLOY	51
6.1.	<i>Với project dạng Ant, Maven</i>	51
6.2.	<i>Với project dạng Msbuild (.NET)</i>	52
7.	HƯỚNG DẪN AUTO DEPLOY TRÊN JENKINS.....	53
7.1.	<i>Auto deploy với project maven, ant</i>	53
7.2.	<i>Auto deploy với project Msbuild.....</i>	55

HƯỚNG DẪN CẤU HÌNH AUTO BUILD và AUTO DEPLOY TRÊN JENKINS

NOTE: Mỗi dự án chú ý:

- Có ba loại dự án chính: Ant, Maven, Msbuild (project .NET). Với mỗi dự án cần **xác định loại dự án** tương ứng và thực hiện cấu hình tương ứng với mỗi loại.
- Mỗi một project tạo một **JOB JENKINS** để thực hiện việc BUILD PROJECT đó.
Chỉ cần tạo 1 JOB để thực hiện cả Auto Build và Auto Deploy cho project JAVA (Ant và Maven). Riêng project MSBUILD (.NET) cần tạo 2 JOB (1 JOB thực hiện BUILD và 1 JOB thực hiện DEPLOY)
- Lưu ý:
 - o Tạo project **MSBUILD (Build .NET)** dạng **Freestyle** và cấu hình **Restrict where this project can be run** tương ứng trên jenkins window slave: **slave_203**
 - o Tạo project **Ant** dạng **Freestyle** và cấu hình **Restrict where this project can be run** tương ứng trên jenkins slave: **slave_43**
 - o Tạo project **Maven** dạng **Maven Project** và cấu hình **Restrict where this project can be run** tương ứng trên jenkins slave: **slave_43**

Các bước cấu hình Auto Build:

Chú ý các bước cấu hình lấy source code trên gitlab/IBM và cấu hình push artifact tới server nexus là tương tự nhau đối với cả ba loại project. (Xem hướng dẫn cấu hình project Ant và làm tương tự với maven và msbuild)

- Repository được tạo trên Nexus để lưu bản build artifact là repo_ant, repo_msbuild, repo_maven. Cấu hình push artifact tới nexus repo tương ứng:
 - o Project Ant: **repo_ant**
 - o Project Maven: **maven-snapshots**
 - o Project Msbuild (.Net): **repo_msbuild**
- Link đường dẫn nội bộ: Nếu chưa có account, liên hệ phòng CNSX để được cấp account.
 - o Jenkins: <http://10.60.156.96:8080/>
 - o Gitlab: <http://10.60.156.11>
 - o Nexus: <http://10.60.156.26:8081>
 - o Jenkins slave:

- ✓ Slave build .NET: `slave_203`
- ✓ Slave build Java (ant và maven) và deploy: `slave_43`

1. HƯỚNG DẪN CẤU HÌNH AUTO BUILD PROJECT DẠNG ANT

Bước 1: Tạo một job tương ứng cho dự án.

Lưu ý: Chọn loại project: **Freestyle**

- Nhập tên Job cần tạo
- Chọn **Freestyle project**
- Click **OK** để hoàn thành tạo job.

Jenkins > All >

Enter an item name

SmSBrandName 1

» Required field

Freestyle project 2

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK 3

Tạo mới project dạng Ant

Bước 2: Cấu hình chung cho job

Thực hiện cấu hình số lượng bản build muốn lưu:

- Tích chọn checkbox **Discard old builds** để cấu hình số lượng bản build muốn giữ lại.
Có thể cấu hình giữ bản build trong 3 ngày và tối đa giữ 10 bản.

General GitBucket Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Define a Docker template

☒ Discard old builds

Strategy: Log Rotation

Days to keep builds: 3
if not empty, build records are only kept up to this number of days

Max # of builds to keep: 10
if not empty, only up to this number of build records are kept

Days to keep artifacts: 3
if not empty, artifacts from builds older than this number of days will be deleted, but the logs, history, reports, etc for the build will be kept

Max # of builds to keep with artifacts: 10
if not empty, only up to this number of builds have their artifacts retained

✚ **Cấu hình tham số sử dụng cho job** - ở đây sử dụng **Tag** trong **Git** để đánh **version** cho bản build. Tương đương với việc job sẽ được Trigger build khi có Tag event → **Phần cấu hình này là tùy chọn**

- Tích chọn **This project is parameterized** → Chọn **Add Parameter** → Chọn **Git Parameter** để lấy các tham số trên gitlab.

Build_eShop

General **GitBucket** Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds

JDK

☒ Restrict where this project can be built

Label Expression

Add Parameter 1

- Boolean Parameter
- Choice Parameter
- Credentials Parameter
- File Parameter
- Git Parameter** 2
- JIRA Issue Parameter
- JIRA Release Version Parameter
- List Subversion tags (and more)
- List maven artifact versions
- Multi-line String Parameter
- Non-Stored Password Parameter
- Password Parameter
- Run Parameter
- String Parameter

Advanced...

- Cấu hình biến cho phù hợp.

✚ **Name:** Đặt tên biến để sử dụng cho các phần cấu hình sau. Ở ví dụ này đặt tên biến là **tag**

✚ **Parameter Type:** Lựa chọn tham số trên git. Có một số lựa chọn như Tag, Branch, Branch or Tag, Revision, Pull Request

✚ **Sort Mode :** Lựa chọn mode để sắp xếp

✚ **Selected Value:** Giá trị Tag sẽ được lựa chọn lấy giá trị trên cùng – TOP

⇒ **Để có thể sử dụng được giá trị chính xác của Tag yêu cầu khi đánh Tag cần đánh version tăng dần.**

✚ Cấu hình môi trường

- Cấu hình JDK được sử dụng cho project: Chọn version JDK cho phù hợp. Ví dụ: JDK7, JDK8,...
- Cấu hình máy chủ thực hiện build. Jenkins cho phép cấu hình chạy job trên các **Node** khác nhau (Jenkins được thiết kế theo mô hình master-slave). Ở đây job được cấu hình chạy trên slaver.
- Trong trường hợp không cấu hình ***Restrict where this project can be run***, Job sẽ được cấu hình mặc định chạy trên máy master.
- Với project dạng Ant không cần cấu hình ***Restrict where this project can be run***.

General **GitBucket** Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Delivery Pipeline configuration

☐ This build requires lockable resources

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

JDK

JDK to be used for this project

☒ Restrict where this project can be run

Label Expression

[Label slave_43](#) is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

Advanced...

Cấu hình chung cho job

Bước 3: Cấu hình đường dẫn tới bản source code

Ở đây, Code được quản lý trên IBM và Gitlab. Đối với từng công nghệ được sử dụng, cấu hình lấy source code tương ứng như sau:

Trong trường hợp code được quản lý trên IBM:

- Thêm account qlsx để có thể lấy source từ dự án: Click link dưới
http://10.61.68.128:8080/credentials/store/system/domain/_/newCredentials
- Nhập username/password đăng nhập qlsx → Click **OK**

Jenkins > Credentials > System > Global credentials (unrestricted)

Back to credential domains

Add Credentials

Kind

Scope

Username

Password

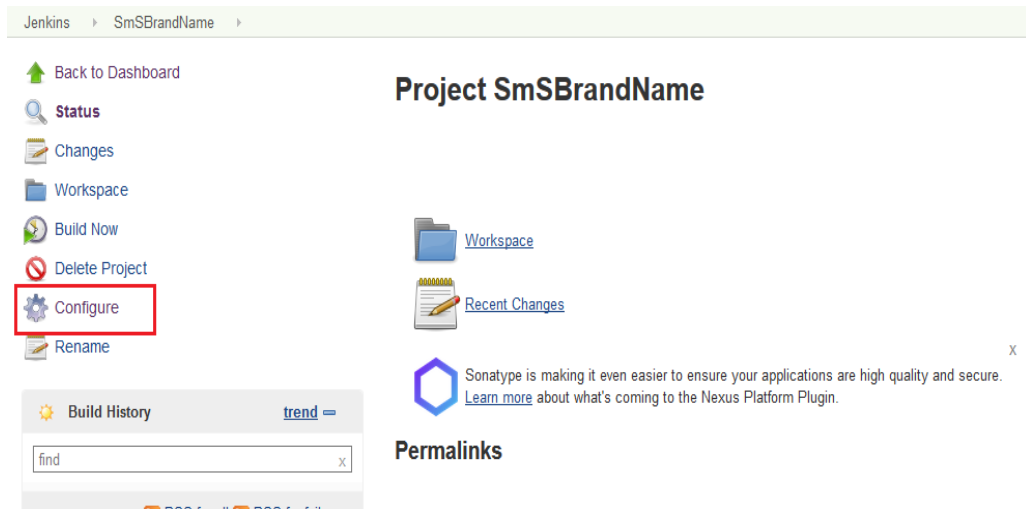
ID

Description

OK

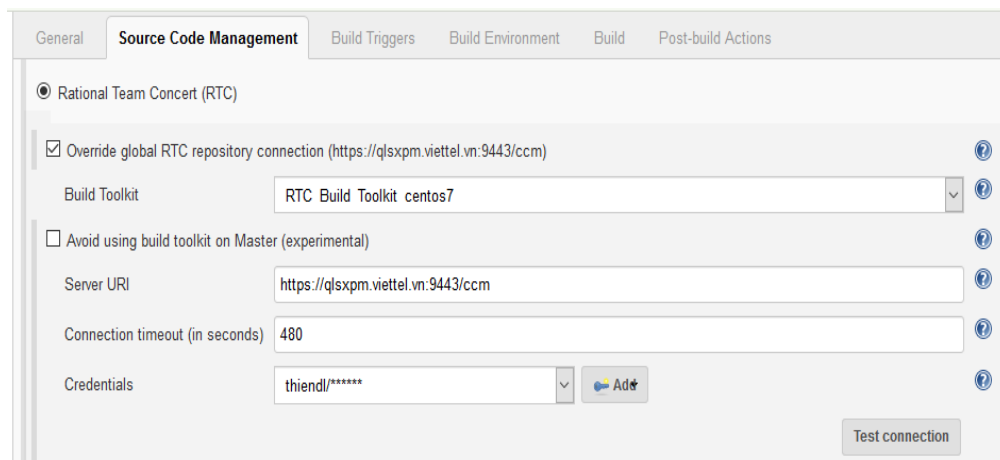
Tạo mới một credentials

- Vào lại job đã tạo ở bước 1 để thực hiện cấu hình:



Thực hiện cấu hình job

Thực hiện cấu hình lấy source từ IBM như dưới (Click radio **Rational Team Concert (RTC)** → Chọn Ccheckbox **Override global RTC repository connection** (<https://qlsxp.m.viettel.vn:9443/ccm>) → chọn RTC trong Build Toolkit → Chọn **Account** đã được cấu hình ở trên.



Cấu hình lấy Source code từ IBM - 1

- Chọn thông tin cần lấy: theo **Stream** hay **Snapshot**,...
- Nhập các thông tin cần thiết như **Tên dự án**, **Tên Stream**, **Tên Snapshot**,...

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Build Configuration Stream

Project or Team Area GPDN_QT06_17011_MULTICAST (Change Management)

Stream SmsBrandname

Snapshot Options

☐ Override default snapshot name

Load Options

Load directory

☒ Delete directory before loading

Load policy Specify which components to load

☒ Create folders for components

Components to load Exclude some components

Components to exclude DOCUMENT

Cấu hình lấy Source code từ IBM – 2

Lưu ý: Cần tích chọn hai options **Delete directory before loading** và **Create folders for components** để việc lấy source code từ IBM không bị lỗi.

Ngoài ra nên **Exclude** một số components không cần thiết để giảm thiểu tài nguyên và thời gian lấy source code.

Trong trường hợp source code được quản lý trên Gitlab:

- Đường dẫn gitlab theo dải nội bộ: **10.60.156.11**

General GitBucket **Source Code Management** Build Triggers Build Environment Build Post-build Actions

☒ Git

Repositories

Repository URL **Link project trên gitlab: 10.60.156.11**

Credentials **username/access token gitlab**

Branches to build

Branch Specifier (blank for 'any') **Branch trên gitlab**

Git executable

Repository browser

Additional Behaviours

Cấu hình lấy Source code từ Gitlab

Lưu ý: Trong trường hợp source code được quản lý trên gitlab, cấu hình thêm build trigger để tự động trigger build. Ví dụ cấu hình như sau:

- Chọn checkbox **Build when a change is pushed to GitLab**. **GitLab webhook URL:** http://10.60.156.96:8080/project/HSSK_BUILD để enable Trigger tới gitlab
- Cấu hình **Enabled GitLab triggers** với các sự kiện như **Push Events**, **Merge Request Events**,...

☒ Build when a change is pushed to GitLab. GitLab webhook URL

Enabled GitLab triggers

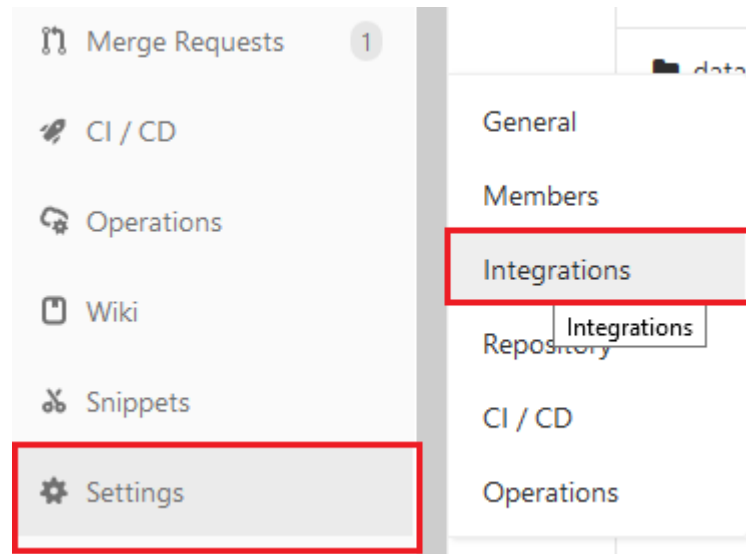
Push Events	<input checked="" type="checkbox"/>
Opened Merge Request Events	<input checked="" type="checkbox"/>
Accepted Merge Request Events	<input type="checkbox"/>
Closed Merge Request Events	<input type="checkbox"/>
Rebuild open Merge Requests	<input type="text" value="Never"/>
Approved Merge Requests (EE-only)	<input checked="" type="checkbox"/>
Comments	<input checked="" type="checkbox"/>
Comment (regex) for triggering a build	<input type="text" value="Jenkins please retry a build"/>

Cấu hình Build Trigger cho job

- Tạo **Secret token** để cấu hình **Add WebHook** trong Gitlab.

Cấu hình WebHook để kích hoạt Trigger Build trong Gitlab

- Đăng nhập tới gitlab: **10.60.156.11**, trong project tương ứng với Job tạo trên Jenkins, vào **Settings** → chọn **Integrations**



Cấu hình Webhook trên gitlab để trigger tới job build trên jenkins - 1

- Cấu hình WebHook để Trigger Build tới Jenkins:

Integrations

Webhooks can be used for binding events when something is happening within the project.

URL	<input type="text" value="http://10.60.156.96:8080/project/HSSK_BUILD"/>
Secret Token	<input type="text" value="b3884e932e87ffefcc1cf0b0a08e0372"/>

Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Token HTTP header.

Cấu hình Webhook trên gitlab để trigger tới job build trên jenkins - 2

Trong đó:

- **URL/Secret Token** là URL/**Secret Token** được lấy trong cấu hình Build Trigger trên Jenkins
- Cấu hình **Trigger** tới Job Build trong Jenkins. Có một số sự kiện để **Trigger** như sau:

Trigger

☒ **Push events**

This URL will be triggered by a push to the repository

Branch name or wildcard pattern to trigger on (leave blank for all)

☐ **Tag push events**

This URL will be triggered when a new tag is pushed to the repository

☐ **Comments**

This URL will be triggered when someone adds a comment

☐ **Confidential Comments**

This URL will be triggered when someone adds a comment on a confidential issue

☐ **Issues events**

This URL will be triggered when an issue is created/updated/merged

☐ **Confidential Issues events**

This URL will be triggered when a confidential issue is created/updated/merged

☐ **Merge request events**

This URL will be triggered when a merge request is created/updated/merged

☐ **Job events**

This URL will be triggered when the job status changes

☐ **Pipeline events**

This URL will be triggered when the pipeline status changes

Một số sự kiện để kích hoạt Trigger

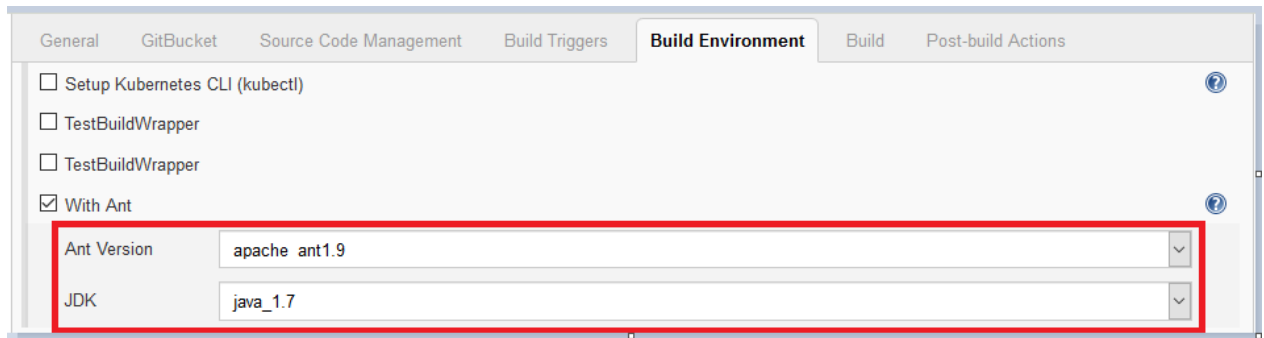
Ví dụ Trigger khi có một số sự kiện phổ biến như sau:

- **Push Events:** Thực hiện Trigger Build khi có sự kiện push tới source code trên gitlab
- **Tag push events:** Thực hiện Trigger Build khi một tag mới được push tới source code.
- **Comments:** Thực hiện Trigger Build khi thêm comment mới.
- **Merge request events:** Thực hiện trigger build khi có một request merge được tạo, cập nhật hay merged.

Bước 4: Cấu hình build job

- **Cấu hình môi trường Build:**

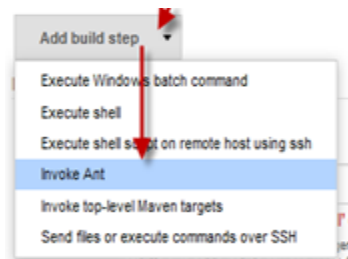
Thực hiện như sau: Kích vào Checkbox **With Ant** → chọn **Ant Version** và **JDK Version** cần sử dụng để build project.



Cấu hình môi trường thực hiện build

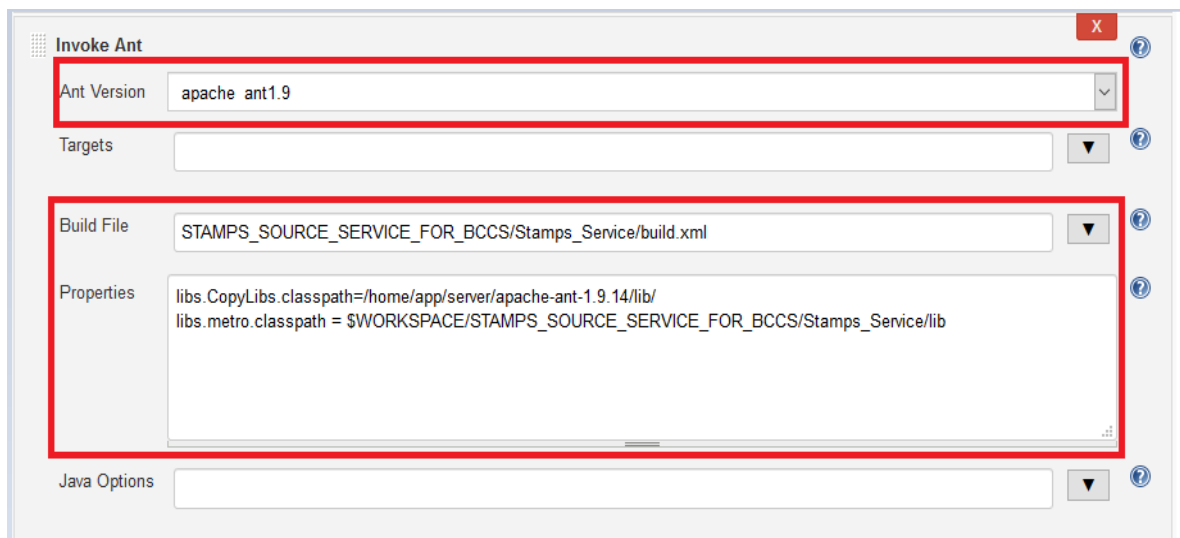
- Cấu hình Build:

- Chọn **Add build Step** → Click **Invoke Ant**



Cấu hình build cho job - 1

- Cấu hình Build cho job như sau:



Cấu hình build cho job - 2

Trong đó:

- ✓ **Ant Version:** Phiên bản Ant dùng để Build Project

- ✓ **Target:** nếu để trống, kết quả Build sẽ nằm trong thư mục mặc định được cấu hình trong project.
- ✓ **Build File:** đường dẫn trỏ tới file ***build.xml***
- ✓ **Properties:** Cho phép cấu hình các thuộc tính đặc biệt được sử dụng riêng cho project. Ví dụ:

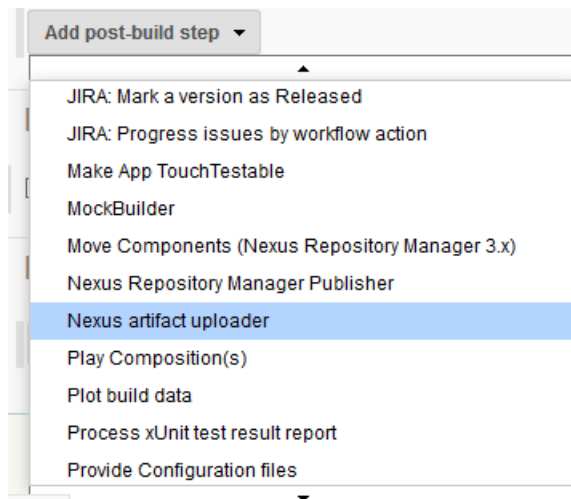
platforms.JDK_1.6.home=/home/cnsx/server/jdk1.7.0_75

libs.CopyLibs.classpath=a/path/to/org-netbeans-modules-java-j2seproject-copylibstask.jar

Bước 5: Cấu hình đẩy artifact tới Nexus Server

Đây là bước cấu hình để đẩy bản build lên server quản lý Build Artifact.

- Chọn ***Nexus artifact uploader*** để cấu hình đẩy file build lên **Nexus Repository Server**.



Cấu hình đẩy bản build tới server Nexus - 1

- Cấu hình các thông tin để có thể đẩy bản build tới server **Nexus** như sau:

Nexus artifact uploader

Nexus Details

Nexus Version: NEXUS3

Protocol: HTTP

Nexus URL: 10.60.156.26:8081

Credentials: admin/***** (nexus) **username/password đăng nhập Nexus server**

GroupId: SMSBrandname

Version: 1.\${BUILD_NUMBER}-CMSCP

Repository: repo_ant

Artifacts

Artifact

ArtifactId: CMSCP

Type: zip

Classifier:

File: SOURCE_CODE/CMS_CP/dist/BulkCP.zip

Add

Cấu hình đẩy bản build tới server Nexus

Trong đó:

- ✓ **Nexus Version:** là version của Nexus server.
- ✓ **Protocol:** Giao thức kết nối HTTP/HTTPS
- ✓ **Nexus URL:** URL tới nexus server
- ✓ **Credentials:** username và password login nexus server repository
- ✓ **GroupID:** tên group do người dùng tự đặt
- ✓ **Version:** tên do người dùng tự đặt để đánh dấu version của bản build.
- ✚ Ví dụ: 1. `1. ${BUILD_NUMBER}-SNAPSHOT` thì version của bản build sẽ được đánh dấu tương ứng với số thứ tự của lần build.
- ✚ Lưu ý: đối với Repository là snapshot cần thêm SNAPSHOT như ví dụ trên.
- ✓ **Repository:** là tên Repository được tạo trong Nexus Repository Server.

- ✓ **ArtifactsId:** tên do người dùng tự đặt.
- ✓ **Type:** loại file sẽ được lưu trên Nexus Repository Server.
- ✓ **File:** đường dẫn đến file kết quả build.

Bước 6: Cấu hình Auto Deploy

- Để cấu hình Auto Deploy, cần viết file Ansible-playbook để mô tả kịch bản các bước cần thực hiện để tiến hành deploy bản build tới server. File Ansible-playbook này có dạng file ***.yaml**.
- Ví dụ đối với project trên, để tiến hành Auto Deploy có thể mô tả một kịch bản chạy Ansible-playbook gồm bốn bước:
 - Stop tiến trình tomcat đang chạy
 - Copy bản build tới server
 - Giải nén bản build tại server
 - Start lại tomcat
- Như vậy file ansible-playbook mô tả các bước deploy như sau:

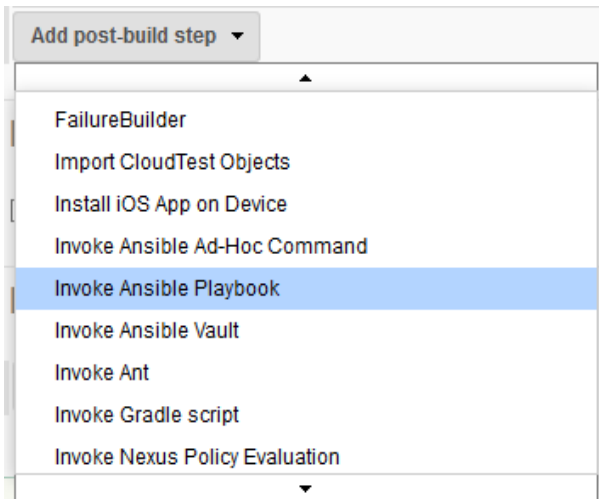
```
- hosts: 10.60.156.71
  remote_user: app
  tasks:
    - name: kill process running
      shell: ps -ef | grep 'tomcat_cms' | grep -v grep | awk '{print $2}' | xargs -r kill -9
    - name: Copy file zip thu mục
      copy:
        src: "dist/BulkCP.zip"
        dest: "/home/app/smsbrandname/tomcat_cms/webapps/BulkCP.zip"
    - name: unzip file BulkCP
      unarchive:
        src: /home/app/smsbrandname/tomcat_cms/webapps/BulkCP.zip
        dest: /home/app/smsbrandname/tomcat_cms/webapps/BulkCP
        remote_src: yes
    - name: start tomcat
      command: chdir=/home/app/smsbrandname/tomcat_cms/bin nohup ./startup.sh
```

Hướng dẫn viết file ansible-playbook để thực hiện deploy một webapp tới server tomcat

Trong đó:

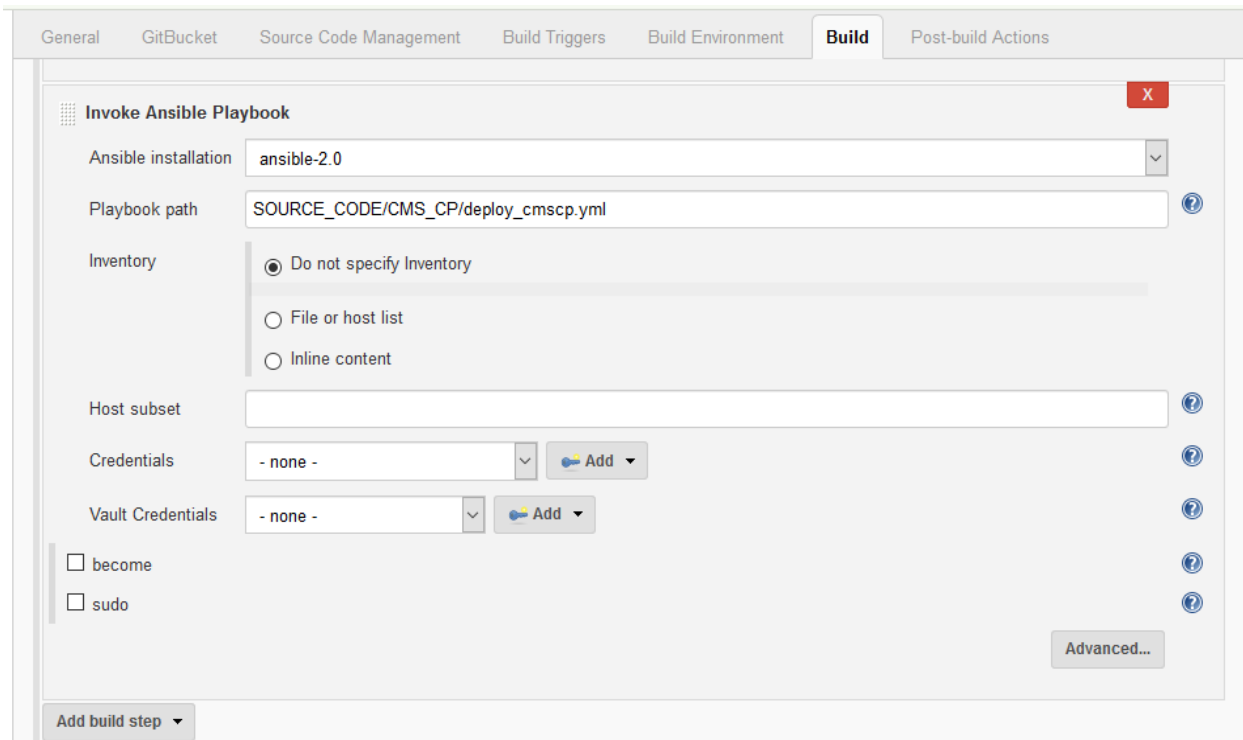
- ✓ **Hosts:** địa chỉ ip hoặc host được cấu hình trong file **/etc/ansible/hosts**.
- ✓ **Tasks:** Mô tả các bước cần thực hiện trên server để deploy bản build.
- ✓ **Name:** tên từng bước thực hiện tương ứng với bốn bước ở trên.

- ✓ **Các module:** gồm tên các module hỗ trợ việc thực hiện các tác vụ như thực hiện việc copy file, chạy command line,...
- Chọn ***Invoke Ansible Playbook*** để cấu hình Auto Deploy cho Job sau khi tạo được một file ansible-playbook.



Cấu hình auto deploy sử dụng ansible - 1

- Cấu hình để chạy Auto Deploy sau khi build xong như dưới đây:



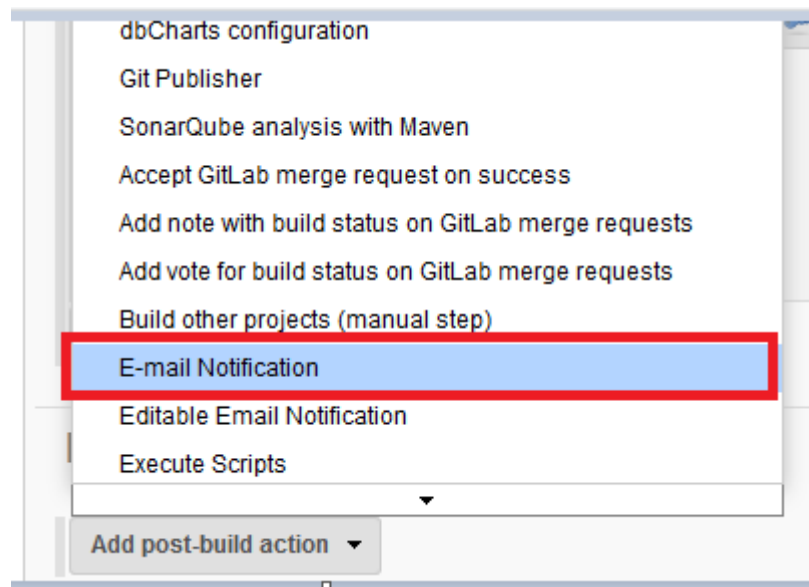
Cấu hình auto deploy sử dụng ansible - 2

Trong đó:

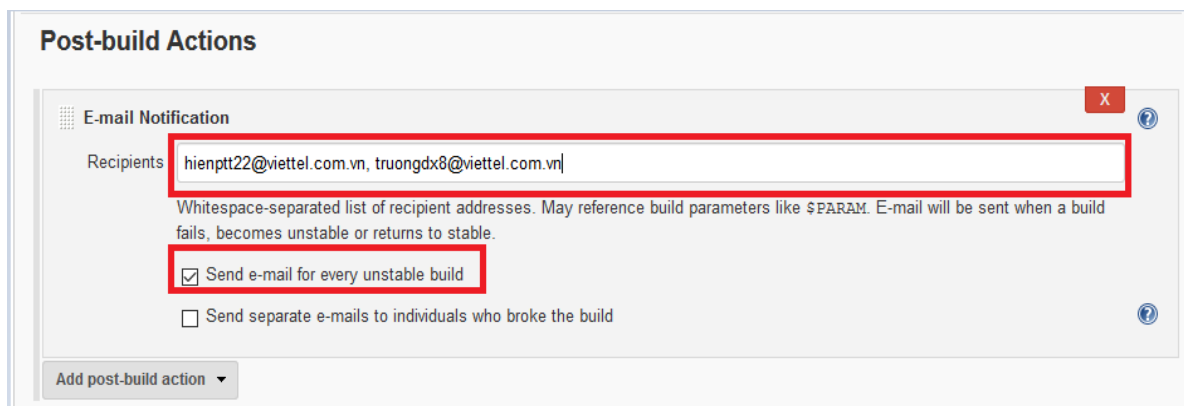
- **Ansible Installation:** Chọn phiên bản Ansible đã được cài đặt trên server.
- **Playbook path:** đường dẫn trỏ đến file ansible play-book trong source code.

Bước 7: Cấu hình gửi email thông báo khi build faild

- Vào **Add post-build action → E-mail Notification:**



- Cấu hình mail của PM – (người nhận thông báo khi build faild):



NOTE: Hướng dẫn trên chỉ gửi mail khi job chạy faild. Trong trường hợp gửi mail bất kì khi nào job chạy success, faild, unstable thì cấu hình như sau:

- Chọn Post-build Actions → Editable Email Notification:

Post-build Actions

Editable Email Notification [X] [?]

Disable Extended Email Publisher ☐ [?]
Allows the user to disable the publisher, while maintaining the settings

Project From

Project Recipient List [?]

- Click **Advance Settings**:

Attach Build Log [?]

Content Token Reference

Click [?]
 [Advanced Settings...]

- Click **Advance**:

Triggers

Always [X] [?]

Send To

- Developers [X] [?]
- Recipient List [X] [?]

Add [v]

Advanced... [?]

Add Trigger [v]

- Nhập danh sách người nhận, cách nhau bởi dấu phẩy, có thể chọn trigger gửi khi failed, success, or always

Recipient List [?]

Reply-To List [?]

Content Type [?]

Subject [?]

Content [?]

Attachments [?]
Can use wildcards like 'module/dist/**/*'.zip'. See the [@includes of Ant files](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log [?]

2. HƯỚNG DẪN CẤU HÌNH AUTO BUILD ĐỐI VỚI PROJECT DẠNG MAVEN

Bước 1: Tạo mới một project tương ứng với dự án.

Chú ý: chọn **Maven Project**

The screenshot shows the Jenkins 'New Item' page. At the top, the Jenkins logo and a search bar are visible. Below the header, the page title is 'Enter an item name'. A text input field contains 'test_maven'. Below the input field, there is a 'Required field' message. The main content area lists four project types: 'Freestyle project', 'Maven project' (which is highlighted with a blue border), 'Pipeline', and 'External Job'. Each project type has a brief description. At the bottom, there is an 'OK' button and a partially visible 'Multi-configuration project' option.

Bước 2: Cấu hình Job

Thực hiện cấu hình số lượng bản build muốn lưu:

- Tích chọn checkbox ***Discard old builds*** để cấu hình số lượng bản build muốn giữ lại. Có thể cấu hình giữ bản build trong 3 ngày và tối đa giữ 10 bản.

General GitBucket Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Define a Docker template

☒ Discard old builds

Strategy: Log Rotation

Days to keep builds: 3
if not empty, build records are only kept up to this number of days

Max # of builds to keep: 10
if not empty, only up to this number of build records are kept

Days to keep artifacts: 3
if not empty, artifacts from builds older than this number of days will be deleted, but the logs, history, reports, etc for the build will be kept

Max # of builds to keep with artifacts: 10
if not empty, only up to this number of builds have their artifacts retained

Cấu hình tham số sử dụng cho job - ở đây sử dụng **Tag** trong **Git** để đánh **version** cho bản build. Tương đương với việc job sẽ được Trigger build khi có Tag event → Phần cấu hình này là tùy chọn

- Tích chọn **This project is parameterized** → Chọn **Add Parameter** → Chọn **Git Parameter** để lấy các tham số trên gitlab.

Build_eShop

General **GitBucket** Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds

JDK

☒ Restrict where this project can be built

Label Expression

Add Parameter 1

- Boolean Parameter
- Choice Parameter
- Credentials Parameter
- File Parameter
- Git Parameter** 2
- JIRA Issue Parameter
- JIRA Release Version Parameter
- List Subversion tags (and more)
- List maven artifact versions
- Multi-line String Parameter
- Non-Stored Password Parameter
- Password Parameter
- Run Parameter
- String Parameter

Advanced...

- Cấu hình biến cho phù hợp.

✚ **Name:** Đặt tên biến để sử dụng cho các phần cấu hình sau. Ở ví dụ này đặt tên biến là **tag**

✚ **Parameter Type:** Lựa chọn tham số trên git. Có một số lựa chọn như Tag, Branch, Branch or Tag, Revision, Pull Request

✚ **Sort Mode :** Lựa chọn mode để sắp xếp

✚ **Selected Value:** Giá trị Tag sẽ được lựa chọn lấy giá trị trên cùng – TOP

⇒ **Để có thể sử dụng được giá trị chính xác của Tag yêu cầu khi đánh Tag cần đánh version tăng dần.**

Cấu hình đường dẫn tới bản code trên gitlab:

Source Code Management

☐ None
☒ Git

Repositories

Repository URL:

Credentials:

Advanced...

Add Repository

Branches to build

Branch Specifier (blank for 'any'):

Branch Specifier (blank for 'any'):

Add Branch

Save Apply (Auto)

- Trong đó:
 - ✓ Repository URL: đường dẫn tới project trên gitlab
 - ✓ Credentials: username và token đăng nhập Gitlab

Trong trường hợp code được quản lý trên IBM, cấu hình lấy code từ IBM như sau:

- Chọn cấu hình như dưới (click radio Rational Team Concert (RTC) → click checkbox Override global RTC repository connection (<https://qlsxpm.viettel.vn:9443/ccm>) → chọn RTC trong build toolkit → chọn account đã cấu hình ở trên

Source Code Management

☐ None
☐ Git
☒ Rational Team Concert (RTC)

☒ Override global RTC repository connection (<https://qlsxpm.viettel.vn:9443/ccm>)

Build Toolkit:

☐ Avoid using build toolkit on Master (experimental)

Server URI:

Connection timeout (in seconds):

Credentials:

Test connection

- Chọn thông tin cần lấy: theo stream hay snapshot...
- Nhập các thông tin cần thiết như tên dự án, tên stream, tên snapshot...

Build Configuration Stream ?

Project or Team Area GPDN_VA_12058_AMI.ONE (Change Management) ?

Stream BUDADN_VA_12058_AMI.ONE_Stream ?

Snapshot Options

☐ Override default snapshot name ?

Load Options

Load directory ?

☒ Delete directory before loading

Load policy Specify which components to load ?

☐ Create folders for components ?

Components to load Exclude some components ?

Components to exclude AMI_MDIS,AMI_MISC,AMI_FIF ?

Save Apply

- **Lưu ý:** Cần tích chọn 2 options *Delete directory before loading* và *Create folders for components* để việc get sourcecode từ IBM không bị lỗi.
- Ngoài ra nên **Exclude** một số components không cần thiết để giảm thiểu tài nguyên và thời gian lấy sourcecode

Cấu hình build job:

- Cấu hình build khi project phụ thuộc build success

Pre Steps

Trigger/call builds on other projects X

Build Triggers

Projects to build mve-common-build ?

☒ Block until the triggered projects finish their builds ?

Fail this build step if the triggered build is worse or equal to FAILURE ?

Mark this build as failure if the triggered build is worse or equal to FAILURE ?

Mark this build as unstable if the triggered build is worse or equal to UNSTABLE ?

Add Parameters ▼

Add ParameterFactories ▼

Add trigger...

- Cấu hình build module:

Build

Maven Version: Apache Maven 3.2.3

Root POM: pom.xml

Goals and options: -Pprod,no-liquibase -Dmaven.test.skip=true -DskipTests clean package

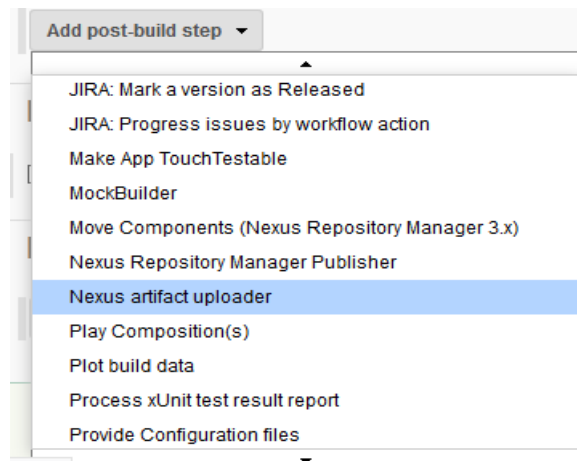
Advanced...

Trong đó:

- **Maven Version:** Phiên bản Maven được sử dụng để build project
- **Root POM:** đường dẫn trỏ tới file pom.xml
- **Goals and options:** cấu hình lệnh chạy để build project ví dụ khi chạy trên máy của lập trình viên cần chạy lệnh *mvn clean install* thì cấu hình *clean install* trong phần này. Tùy từng project, nội dung cấu hình có thể khác nhau. Ví dụ cấu hình build job bằng cách chạy lệnh: *-Pprod,no-liquibase -Dmaven.test.skip=true -DskipTests clean package*.

Bước 3: Cấu hình push artifact tới nexus server

- Chọn *Nexus artifact uploader* để cấu hình đẩy file build lên nexus repository server



- Cấu hình thông tin để đẩy file build lên nexus như sau:

Nexus Version: NEXUS3

Protocol: HTTP

Nexus URL: 10.60.156.26:8081

Credentials: admin/***** (nexus) Add

GroupId: test

Version: 1.\${BUILD_NUMBER}-SNAPSHOT

Repository: maven-snapshots

Artifacts

Artifact

ArtifactId: tomcat-test

Type: war

Classifier:

File: target/tomcat-test-1.0-SNAPSHOT.war

Save Apply

Bước 4: Chạy Job và check kết quả

- Kết quả build job:

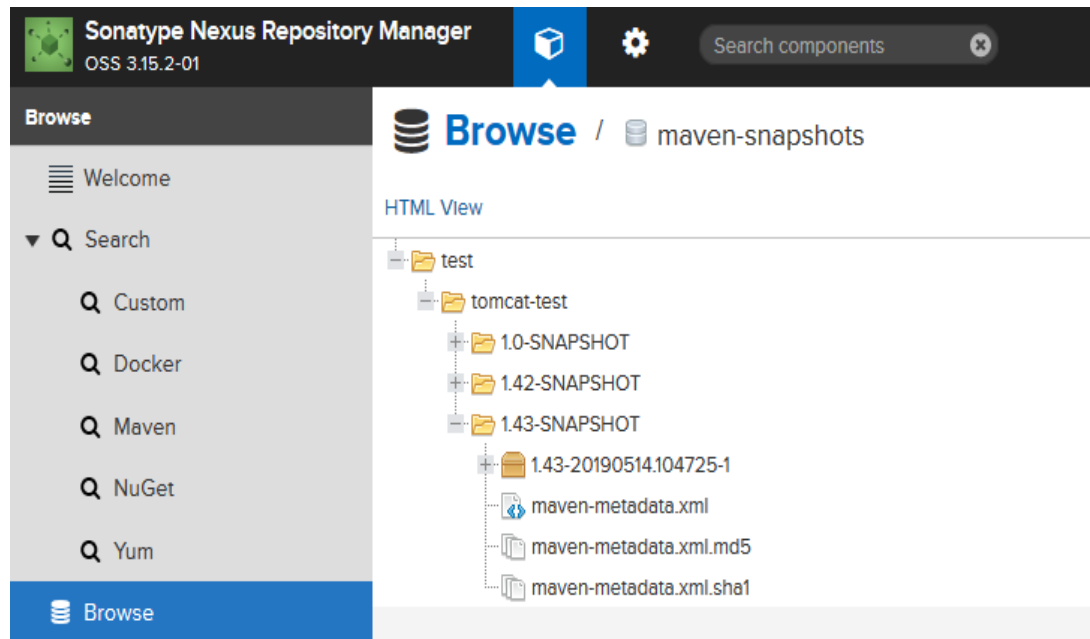
```
Jenkins > test_maven > #36

[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ tomcat-test ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/app/.jenkins/workspace/test_maven/src/test/resources
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ tomcat-test ---
[INFO] No sources to compile
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ tomcat-test ---
[INFO] No tests to run.
[WARNING] Attempt to (de-)serialize anonymous class hudson.maven.reporters.BuildInfoRecorder$1; see: https://jenkins.io/redirect/serialization-of-anonymous-classes/
[INFO] --- maven-war-plugin:2.3:war (default-war) @ tomcat-test ---
[INFO] Packaging webapp
[INFO] Assembling webapp [tomcat-test] in [/home/app/.jenkins/workspace/test_maven/target/tomcat-test-1.0-SNAPSHOT]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/app/.jenkins/workspace/test_maven/src/main/webapp]
[INFO] Webapp assembled in [21 msecs]
[INFO] Building war: /home/app/.jenkins/workspace/test_maven/target/tomcat-test-1.0-SNAPSHOT.war
[WARNING] Attempt to (de-)serialize anonymous class hudson.maven.reporters.MavenArtifactArchiver$2; see: https://jenkins.io/redirect/serialization-of-anonymous-classes/
[WARNING] Attempt to (de-)serialize anonymous class hudson.maven.reporters.MavenFingerprinter$1; see: https://jenkins.io/redirect/serialization-of-anonymous-classes/
[INFO] BUILD SUCCESS
[INFO] Total time: 2.758 s
[INFO] Finished at: 2019-05-10T15:29:14+07:00
[INFO] Final Memory: 17M/208M
```

- Kết quả archive artifacts tới server nexus:

```
[JENKINS] Archiving /home/app/.jenkins/workspace/test_maven/pom.xml to test/tomcat-test/1.0-SNAPSHOT/tomcat-test-1.0-SNAPSHOT.pom
[JENKINS] Archiving /home/app/.jenkins/workspace/test_maven/target/tomcat-test-1.0-SNAPSHOT.war to test/tomcat-test/1.0-SNAPSHOT/tomcat-test-1.0-SNAPSHOT.war
Uploading artifact tomcat-test-1.0-SNAPSHOT.war started...
GroupId: test
ArtifactId: tomcat-test
Classifier:
Type: war
Version: 1.43-SNAPSHOT
File: tomcat-test-1.0-SNAPSHOT.war
Repository:maven-snapshots
channel stopped
Downloading: http://10.60.156.26:8081/repository/maven-snapshots/test/tomcat-test/1.43-SNAPSHOT/maven-metadata.xml
Uploading: http://10.60.156.26:8081/repository/maven-snapshots/test/tomcat-test/1.43-SNAPSHOT/tomcat-test-1.43-20190514.104725-1.war
100 % completed (2.3 kB / 2.3 kB) .
Uploaded: http://10.60.156.26:8081/repository/maven-snapshots/test/tomcat-test/1.43-SNAPSHOT/tomcat-test-1.43-20190514.104725-1.war (2.3 kB at 64 kB/s)
Uploading: http://10.60.156.26:8081/repository/maven-snapshots/test/tomcat-test/1.43-SNAPSHOT/maven-metadata.xml
Uploaded: http://10.60.156.26:8081/repository/maven-snapshots/test/tomcat-test/1.43-SNAPSHOT/maven-metadata.xml (590 B at 30 kB/s)
Uploading artifact tomcat-test-1.0-SNAPSHOT.war completed.
Finished: SUCCESS
```

- Kết quả trên nexus repository server:

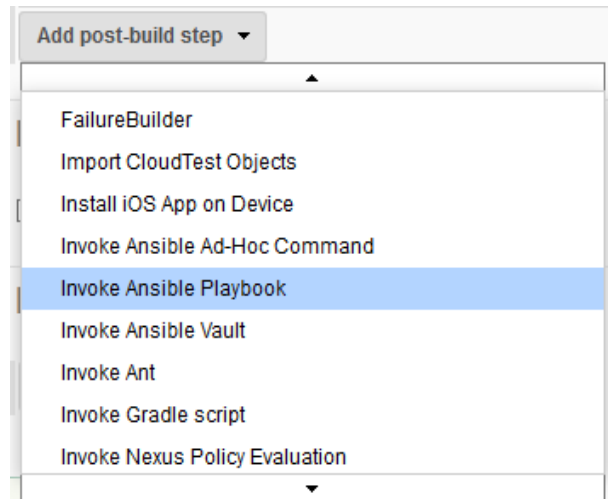


Bước 5: Cấu hình auto deploy sau khi build

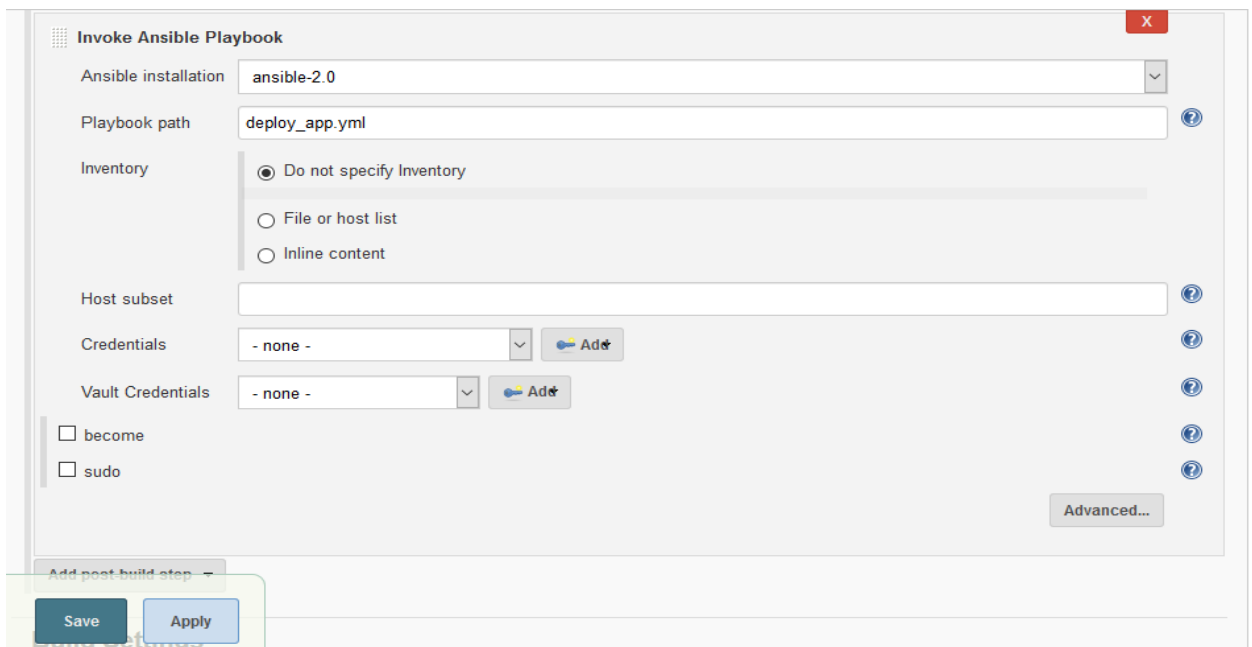
Sử dụng **ansible-playbook** để thực hiện việc auto deploy cho project dạng maven, msbuild.

Ansible được dùng để cấu hình tự động việc kết nối tới server deploy.

Hướng dẫn viết file Ansible playbook mục 4



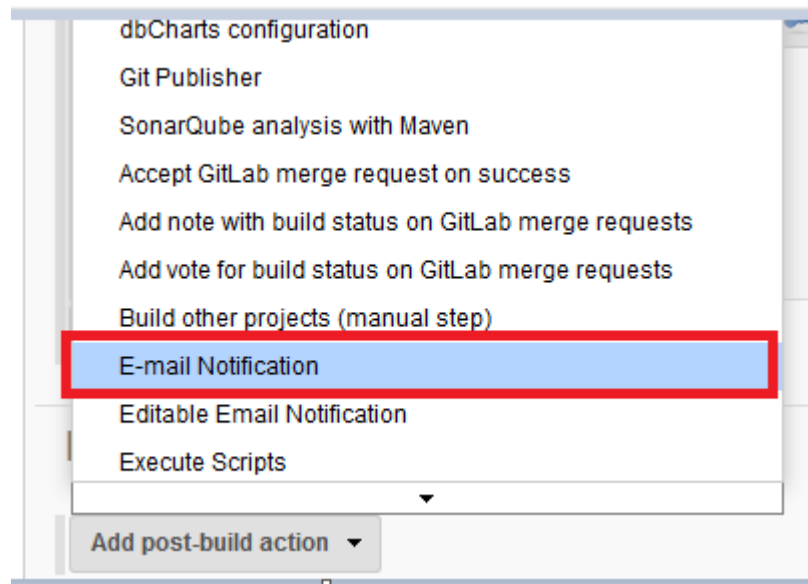
- Thực hiện cấu hình như sau:



- Trong đó:
 - o Ansible installation: phiên bản ansible được cấu hình trong jenkins
 - o Playbook path: cấu hình đường dẫn tới file playbook ansible (file dạng .yaml)

Bước 6: Cấu hình gửi email thông báo khi build faild

- Vào **Add post-build action → E-mail Notification:**



- Cấu hình mail của PM – (người nhận thông báo khi build faild):

Post-build Actions

E-mail Notification [X] [?]

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build

[?] [?]

Add post-build action ▾

NOTE: Hướng dẫn trên chỉ gửi mail khi job chạy faild. Trong trường hợp gửi mail bất kì khi nào job chạy success, faild, unstable thì cấu hình như sau:

- Chọn Post-build Actions → Editable Email Notification:

Post-build Actions

Editable Email Notification [X] [?]

Disable Extended Email Publisher ☐

Allows the user to disable the publisher, while maintaining the settings

Project From

Project Recipient List


[?] [?]

- Click **Advance Settings**:

directory is [the workspace](#).

Attach Build Log Attach Build Log

Content Token Reference

Click 

Advanced Settings...

- Click **Advance**:

Triggers

Always

Send To

Developers

Recipient List

Add

Advanced...

Add Trigger

- Nhập danh sách người nhận, cách nhau bởi dấu phẩy, có thể chọn trigger gửi khi failed, success, or always

Recipient List ducnc2@viettel.com.vn, hienptt22@viettel.com.vn

Reply-To List \$PROJECT_DEFAULT_REPLYTO

Content Type Both HTML and Plain Text

Subject \$PROJECT_DEFAULT_SUBJECT

Content \$PROJECT_DEFAULT_CONTENT

Attachments

Can use wildcards like 'module/dist/**/*.zip'. See the [@includes of Ant fileset](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log Attach Build Log

3. HƯỚNG DẪN CẤU HÌNH AUTO BUILD PROJECT DẠNG MSBUILD

Sử dụng để build project .NET


Bước 1: Tạo một project tương ứng với dự án


Chọn project dạng Freestyle


Enter an item name


msbuild_project


» Required field


Freestyle project
 This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.


Maven project
 Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.


Pipeline
 Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.


External Job
 This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.


Multi-configuration project
 Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

Gitbucket Team/Project

Bước 2: Cấu hình job

- Thực hiện cấu hình số lượng bản build muốn lưu:
- Tích chọn checkbox **Discard old builds** để cấu hình số lượng bản build muốn giữ lại. Có thể cấu hình giữ bản build trong 3 ngày và tối đa giữ 10 bản.

General GitBucket Source Code Management Build Triggers Build Environment Build Post-build Actions

☐ Define a Docker template

☒ Discard old builds

Strategy Log Rotation

Days to keep builds 3

if not empty, build records are only kept up to this number of days

Max # of builds to keep 10

if not empty, only up to this number of build records are kept

Days to keep artifacts 3

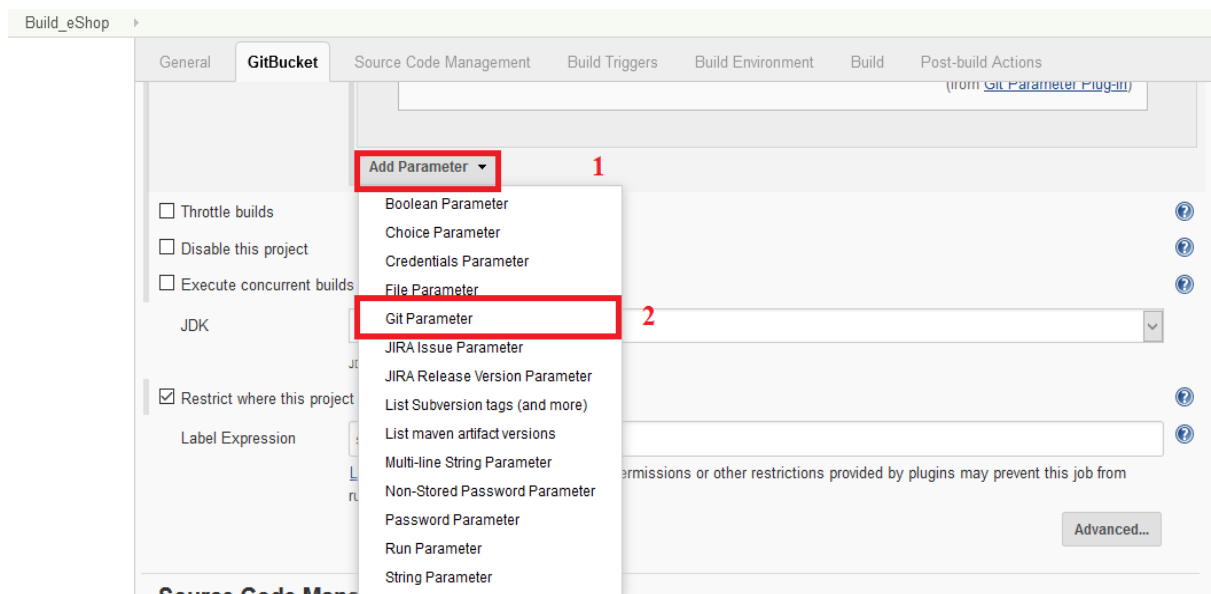
if not empty, artifacts from builds older than this number of days will be deleted, but the logs, history, reports, etc for the build will be kept

Max # of builds to keep with artifacts 10

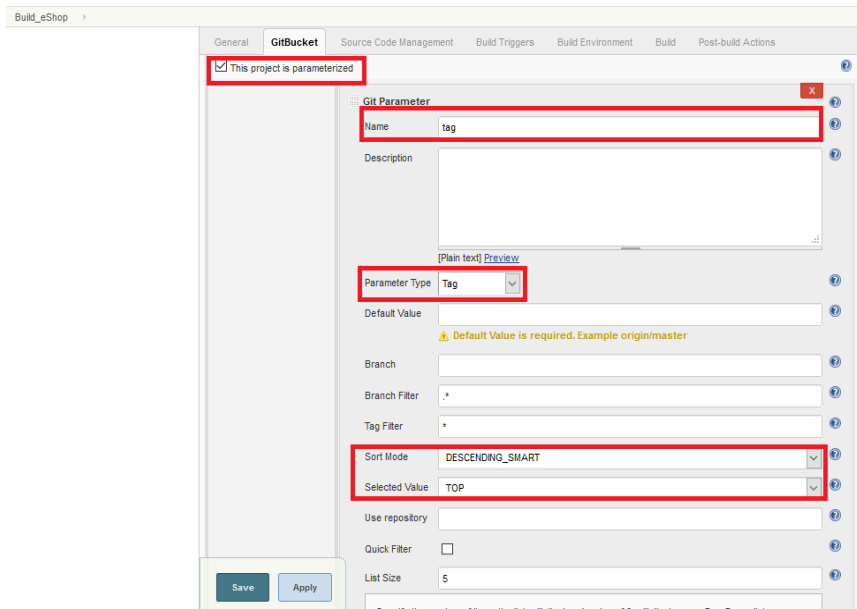
if not empty, only up to this number of builds have their artifacts retained

o Cầu hình tham số sử dụng cho job - ở đây sử dụng **Tag** trong **Git** để đánh **version** cho bản build. Tương đương với việc job sẽ được Trigger build khi có Tag event → **Phần cấu hình này là tùy chọn**

- Tích chọn **This project is parameterized** → Chọn **Add Parameter** → Chọn **Git Parameter** để lấy các tham số trên gitlab.

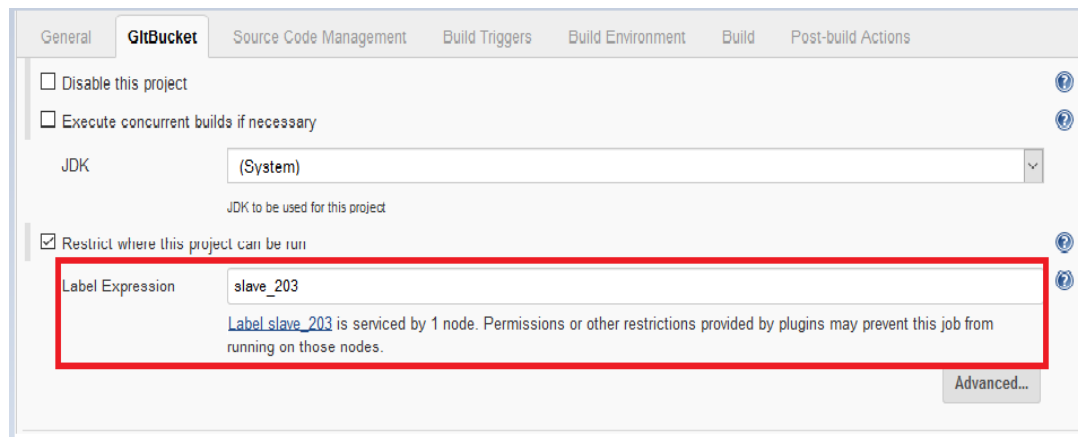


- Cấu hình biến cho phù hợp.
 - + **Name:** Đặt tên biến để sử dụng cho các phần cấu hình sau. Ở ví dụ này đặt tên biến là **tag**
 - + **Parameter Type:** Lựa chọn tham số trên git. Có một số lựa chọn như Tag, Branch, Branch or Tag, Revision, Pull Request
 - + **Sort Mode :** Lựa chọn mode để sắp xếp
 - + **Selected Value:** Giá trị Tag sẽ được lựa chọn lấy giá trị trên cùng – TOP
- ⇒ **Để có thể sử dụng được giá trị chính xác của Tag yêu cầu khi đánh Tag cần đánh version tăng dần.**



○ Cấu hình môi trường

Cấu hình máy chủ thực hiện build. Project Msbuild chỉ build thành công trên môi trường window vì vậy cần cấu hình Jenkins slave window server để build project:



○ Cấu hình build

Một số project cần phải chạy restore để nuget và dotnet để build như sau:

D:\setup\nuget.exe restore "%WORKSPACE%\eShopOnWeb.sln"

"C:\Program Files\dotnet\dotnet.exe" restore "%WORKSPACE%\eShopOnWeb.sln"

Execute Windows batch command

Command

```
D:\setup\nuget.exe restore "%WORKSPACE%\eShopOnWeb.sln"
"C:\Program Files\dotnet\dotnet.exe" restore "%WORKSPACE%\eShopOnWeb.sln"
```

See [the list of available environment variables](#)

Advanced...

Chọn *Build a Visual Studio project or solution using MSBuild* để build project dạng Msbuild.

Build

Add build step ▾

- Conditional step (single)
- Add a new template to all docker clouds
- Ansible Tower
- Build / Publish Docker Image
- Build a Visual Studio project or solution using MSBuild**
- Capture Environment Variables
- Capture iOS Device Screen Shot
- Conditional steps (multiple)
- Copy a file from the workspace of another build
- Create a file
- Deploy to Azure Container Service (AKS)

'configure# ▾

Page generated: May 9, 2019 4:32:51 PM ICT

○ Cấu hình Build job:

Build a Visual Studio project or solution using MSBuild

MSBuild Version (Default) ▾

MSBuild Build File `${WORKSPACE}\eShopOnWeb.sln` ?

Command Line Arguments `/T:Clean;Build /p:Configuration=Release /p:DeployOnBuild=true|` ?

Pass build variables as properties ☐

Do not use chcp command ☐

Advanced...

- Trong đó:
 - ✓ **MSBuild version:** chọn version MSBuild phù hợp.
 - ✓ **MSBuild Build File:** trỏ đến file *.csproj hoặc *.Sln
 - ✓ **Command Line Arguments:** cấu hình build */T:Clean;Build /p:Configuration=Release /p:DeployOnBuild=true* – cấu hình push bản build tới thư mục Publish
- **Cấu hình zip file kết quả build để đẩy lên server nexus:**



Lưu ý: Trong một số project, chỉ copy lên server test một số file mà không copy tất cả các file kết quả build. Có thể xóa file không cần copy lên server trước khi nén lại kết quả như sau:

- **Xóa một file dùng command line trong window:**

del

%WORKSPACE%\VTICT.HISONE.SYSTEM\HisOne.WCFService\obj\Debug\Package\PackageTmp\bin\Oracle.DataAccess.dll

- **Nén file:**

"C:\Program Files\7-Zip\7Z.exe" a -tzip %WORKSPACE%\push.zip %WORKSPACE%\TCMR_Web\Web\obj\Release\Package\PackageTmp*

Bước 3: Cấu hình push artifact tới nexus server

Nexus Details

Nexus Version: NEXUS3

Protocol: HTTP

Nexus URL: 10.60.156.26:8081

Credentials: admin/***** (nexus) Add

GroupId: \${JOB_NAME}

Version: \${tag}

Repository: msbuild

Artifacts

Artifact

ArtifactId: \${JOB_NAME}

Type: zip

Classifier:

File: \${WORKSPACE}\eshop.zip

Có thể cấu hình bản build artifact với tên của project như trên.

Bước 4: Chạy job và check kết quả

Kết quả lấy source code từ gitlab:

```
using credential 63265de3-8396-40f9-803e-5cd0b694e519
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url http://10.60.156.11/thuhien.tb.96@gmail.com/test.git # timeout=10
Fetching upstream changes from http://10.60.156.11/thuhien.tb.96@gmail.com/test.git
> git --version # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --tags --progress http://10.60.156.11/thuhien.tb.96@gmail.com/test.git +refs/heads/*:refs/remotes/origin/*
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
> git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision c18c1730a7f8d06f023dea03b29ecf2ba0273e4d (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f c18c1730a7f8d06f023dea03b29ecf2ba0273e4d
Commit message: "Update test.yml"
> git rev-list --no-walk c18c1730a7f8d06f023dea03b29ecf2ba0273e4d # timeout=10
```

Kết quả build project trên Jenkins slave:

```

Jenkins > eshop > Build_eShop > #30
Copying file from "C:\Users\app\.nuget\packages\xunit.runner.visualstudio\2.4.1\build\netcoreapp1.0\xunit.runner.visualstudio.dotnetcore.testadapter.dll" to "bin\Release\netcoreapp2.2\xunit.runner.visualstudio.dotnetcore.testadapter.dll".
Copying file from "C:\Users\app\.nuget\packages\xunit.runner.visualstudio\2.4.1\build\netcoreapp1.0\xunit.runner.reporters.netcoreapp10.dll" to "bin\Release\netcoreapp2.2\xunit.runner.reporters.netcoreapp10.dll".
Copying file from "D:\Jenkins\workspace\Build_eShop\src\Web\obj\Release\netcoreapp2.2\Web.Views.dll" to "bin\Release\netcoreapp2.2\Web.Views.dll".
Copying file from "D:\Jenkins\workspace\Build_eShop\src\Web\obj\Release\netcoreapp2.2\Web.Views.pdb" to "bin\Release\netcoreapp2.2\Web.Views.pdb".
Copying file from "C:\Users\app\.nuget\packages\xunit.runner.visualstudio\2.4.1\build\netcoreapp1.0\xunit.runner.utility.netcoreapp10.dll" to "bin\Release\netcoreapp2.2\xunit.runner.utility.netcoreapp10.dll".
_CopyOutOfDateSourceItemsToOutputDirectoryAlways:
Copying file from "D:\Jenkins\workspace\Build_eShop\src\Web\appsettings.json" to "bin\Release\netcoreapp2.2\appsettings.json".
CopyFilesToOutputDirectory:
Copying file from "obj\Release\netcoreapp2.2\FunctionalTests.dll" to "bin\Release\netcoreapp2.2\FunctionalTests.dll".
FunctionalTests -> D:\Jenkins\workspace\Build_eShop\tests\FunctionalTests\bin\Release\netcoreapp2.2\FunctionalTests.dll
Copying file from "obj\Release\netcoreapp2.2\FunctionalTests.pdb" to "bin\Release\netcoreapp2.2\FunctionalTests.pdb".
CopyAdditionalFiles:
Copying file from "D:\Jenkins\workspace\Build_eShop\src\Web\bin\Release\netcoreapp2.2\Web.deps.json" to "bin\Release\netcoreapp2.2\Web.deps.json".
Done Building Project "D:\Jenkins\workspace\Build_eShop\tests\FunctionalTests\FunctionalTests.csproj" (default targets).
Done Building Project "D:\Jenkins\workspace\Build_eShop\eshopOnWeb.sln" (Clean/Build target(s)).

Build succeeded.
    0 Warning(s)
    0 Error(s)

```

Kết quả push artifact tới nexus với version bản build là Tag:

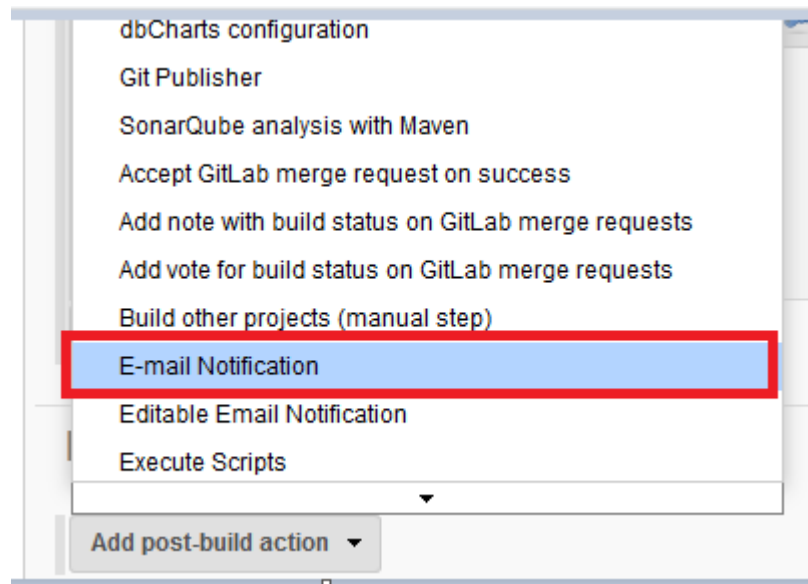
```

E:\Jenkins\workspace\Build_eShop>exit 0
Uploading artifact eshop.zip started....
GroupId: Build_eShop
ArtifactId: Build_eShop
Classifier:
Type: zip
Version: 2.0.1
File: eshop.zip
Repository:msbuild
Uploading: http://10.60.156.26:8081/repository/msbuild/Build_eShop/Build_eShop/2.0.1/Build_eShop-2.0.1.zip
10 % completed (872 kB / 8.7 MB).
20 % completed (1.7 MB / 8.7 MB).
30 % completed (2.6 MB / 8.7 MB).
40 % completed (3.5 MB / 8.7 MB).
50 % completed (4.4 MB / 8.7 MB).
60 % completed (5.2 MB / 8.7 MB).
70 % completed (6.1 MB / 8.7 MB).
80 % completed (7.0 MB / 8.7 MB).
90 % completed (7.9 MB / 8.7 MB).
100 % completed (8.7 MB / 8.7 MB).
Uploaded: http://10.60.156.26:8081/repository/msbuild/Build_eShop/Build_eShop/2.0.1/Build_eShop-2.0.1.zip 8.7 MB at 31 MB/s
Uploading artifact eshop.zip completed.
[InfluxDB Plugin] Collecting data...
[InfluxDB Plugin] Publishing data to target 'InfluxDB' (url='http://10.60.156.104:8086', database='jenkins_db')
[InfluxDB Plugin] Completed.
[WS-CLEANUP] Deleting project workspace...
[WS-CLEANUP] Deferred wipeout is used...
[WS-CLEANUP] done
Finished: SUCCESS

```

Bước 5: Cấu hình gửi email thông báo khi build faild

- Vào Add post-build action → E-mail Notification:



- Cấu hình mail của PM – (người nhận thông báo khi build faild):

Post-build Actions

E-mail Notification X ?

Recipients hienptt22@viettel.com.vn, trungdx8@viettel.com.vn

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build ?

Add post-build action ▼

NOTE: Hướng dẫn trên chỉ gửi mail khi job chạy faild. Trong trường hợp gửi mail bất kì khi nào job chạy success, faild, unstable thì cấu hình như sau:

- Chọn Post-build Actions → Editable Email Notification:

Post-build Actions

Editable Email Notification X ?

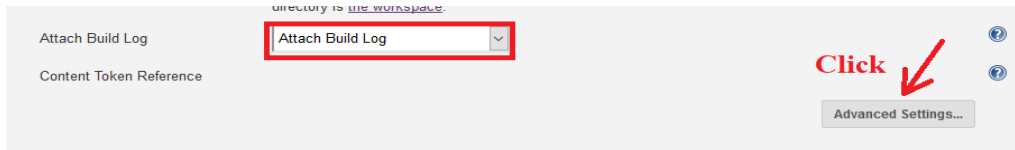
Disable Extended Email Publisher ☐ ?

Allows the user to disable the publisher, while maintaining the settings

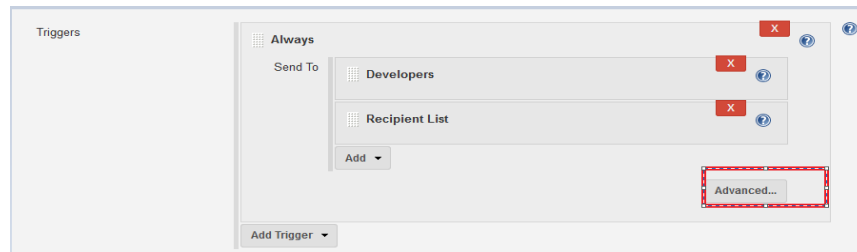
Project From

Project Recipient List \$DEFAULT_RECIPIENTS ?

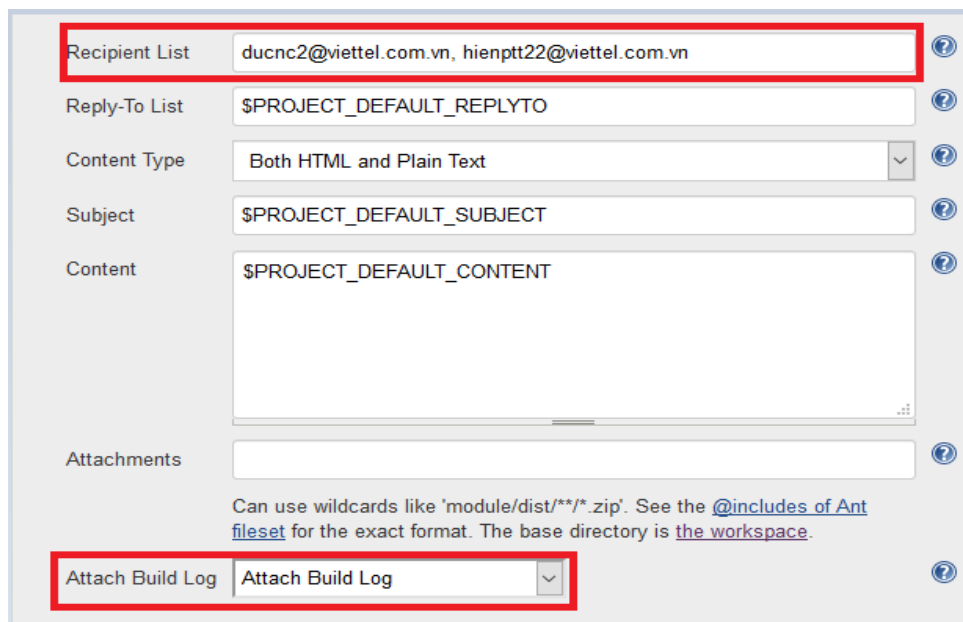
- Click **Advance Settings**:



- Click **Advance**:

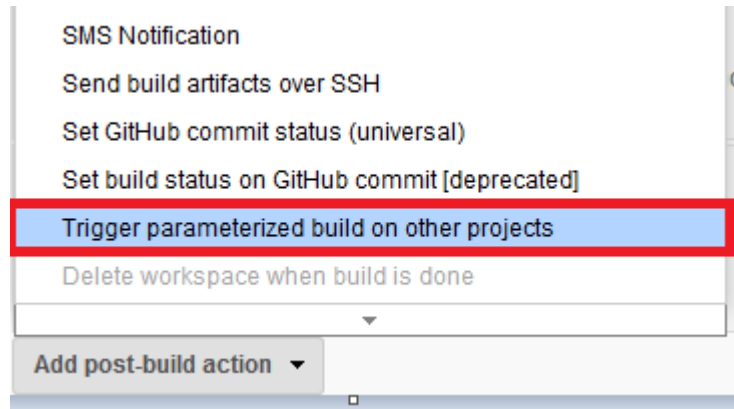


- Nhập danh sách người nhận, cách nhau bởi dấu phẩy, có thể chọn trigger gửi khi failed, success, or always

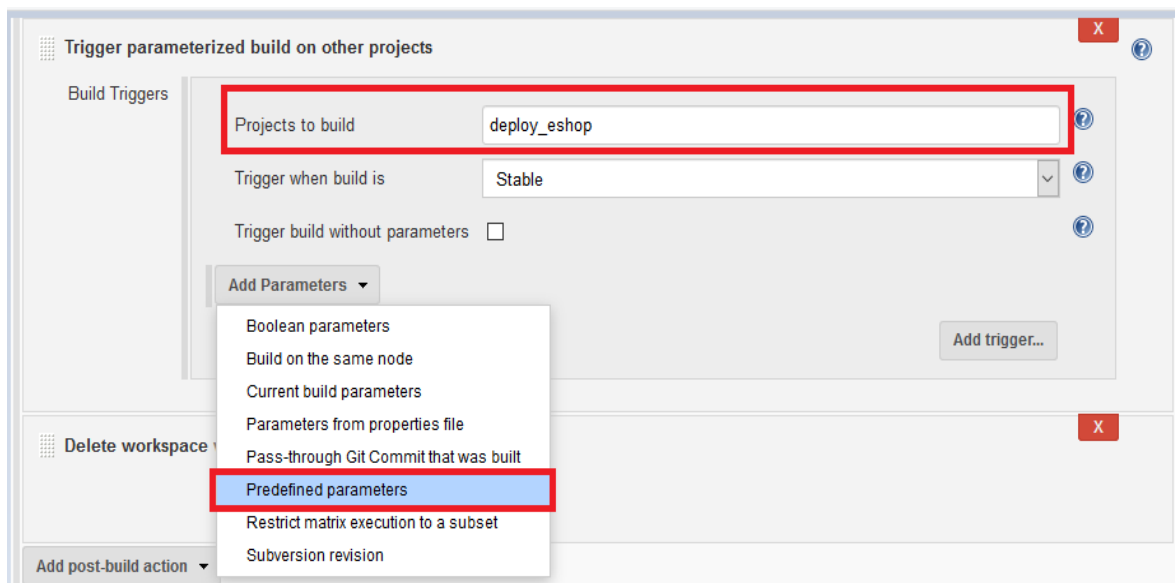


Bước 6: Cấu hình tham số sử dụng cho job thực hiện Auto Deploy:

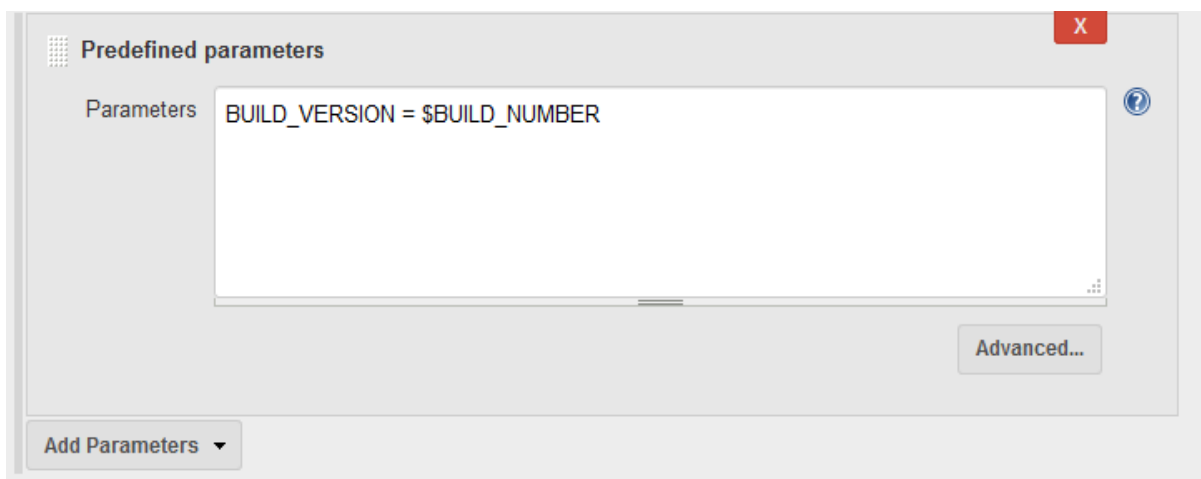
- Cấu hình Trigger chạy Job thực hiện Auto Deploy sau khi Build thành công như sau: Trong **Post-build Actions**, Vào **Add Post-build Actions** → **Trigger parameterized build on other project**.



- Điền tên job thực hiện Auto Deploy và thêm tham số sử dụng cho JOB:



- Ví dụ Job thực hiện Deploy sử dụng BUILD_NUMBER của job Build để lấy version bản build, thực hiện cấu hình như sau:



MỘT SỐ THAM SỐ HAY DÙNG:

- **\$WORKSPACE:** tham số thể hiện đường dẫn tới thư mục workspace

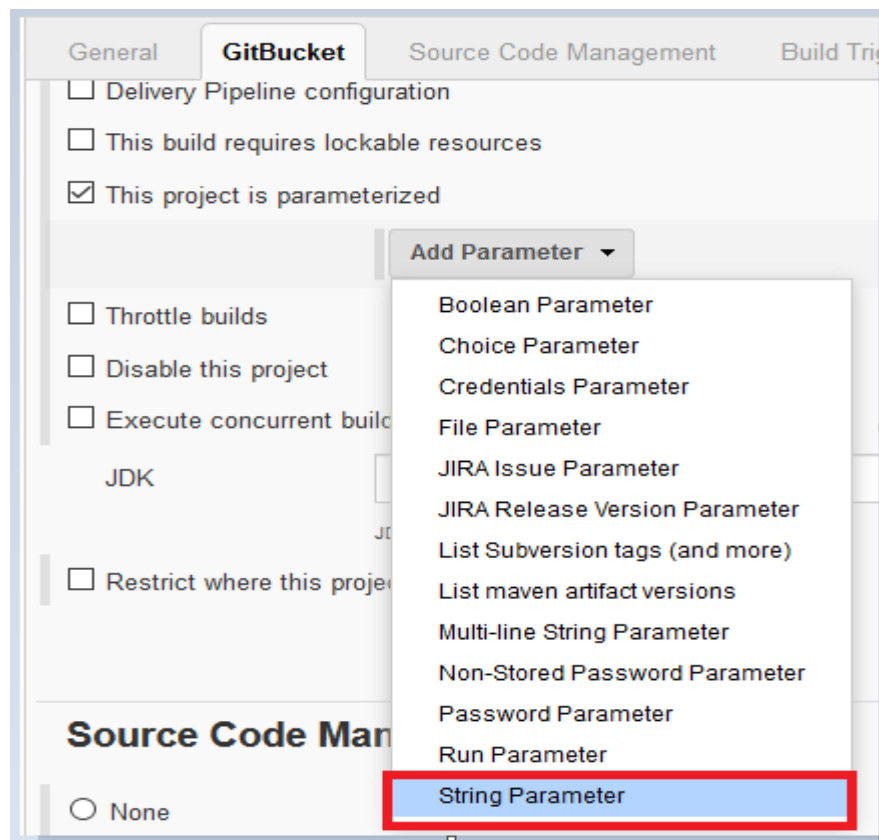
Ví dụ source code được lưu trong thư mục: *D:\Jenkins\workspace\Build_eShop* thì giá trị của **\$WORKSPACE** = *D:\Jenkins\workspace\Build_eShop*

- **\$BUILD_NUMBER:** tham số thể hiện số lần build.
- **\$JOB_NAME:** tên job

Bước 7: Cấu hình job thực hiện Auto Deploy

TRONG JOB THỰC HIỆN AUTO_DEPLOY THỰC HIỆN CẤU HÌNH NHƯ SAU ĐỂ CÓ THỂ CHẠY SỬ DỤNG THAM SỐ TỪ JOB BUILD.

- Trong job thực hiện Auto Deploy, thực hiện cấu hình tham số sử dụng cho job như sau:



☒ This project is parameterized

String Parameter

Name

BUILD_VERSION

Default Value

Description

[Plain text] [Preview](#)

☐ Trim the string

Add Parameter ▾

- **Cấu hình slave chạy Auto Deploy:**

General

GitBucket

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

☐ Disable this project
 ☐ Execute concurrent builds if necessary

JDK

(System)

JDK to be used for this project

☒ Restrict where this project can be run

Label Expression

slave_43

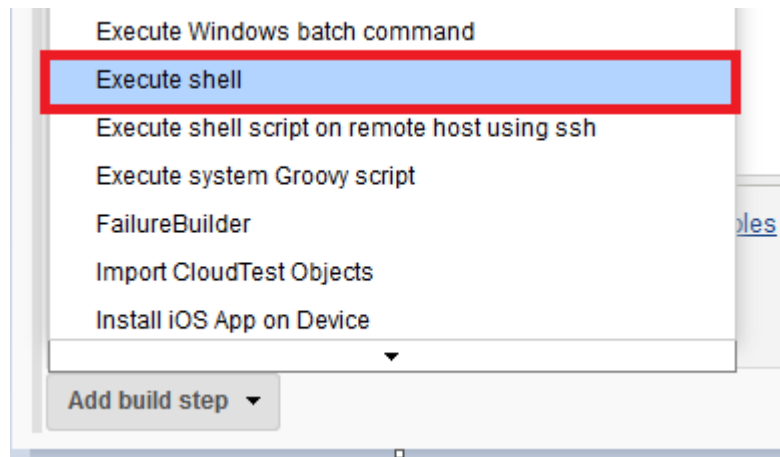
[Label slave_43](#) is serviced by 1 node. Permissions or other restrictions provided by plugins may prevent this job from running on those nodes.

Advanced...

- **Cấu hình đường dẫn tới source:** **Tương tự job build**. Lưu ý, file Ansible-playbook để thực hiện Auto Deploy phải nằm trong thư mục source code.

- **Cấu hình chạy Auto Deploy như sau:**

Vào *Add build step* → *Execute shell* :



4. HƯỚNG DẪN CẤU HÌNH BUILD VỚI PROJECT ANGULAR

Bước 1: Tạo một project tương ứng với dự án

- Chọn project dạng **Freestyle project**

Enter an item name

build_angular 1


* Required field

 **Freestyle project** 2


This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **Maven project**


Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 **Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **External Job**

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

 **Multi-configuration project**

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK 3

Project Name/Project

Bước 2: Cấu hình job

Thực hiện cấu hình số lượng bản build muốn lưu:

- Tích chọn checkbox **Discard old builds** để cấu hình số lượng bản build muốn giữ lại. Có thể cấu hình giữ bản build trong 3 ngày và tối đa giữ 10 bản.

General | GitBucket | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

☐ Define a Docker template

☒ Discard old builds ?

Strategy

Log Rotation

Days to keep builds if not empty, build records are only kept up to this number of days

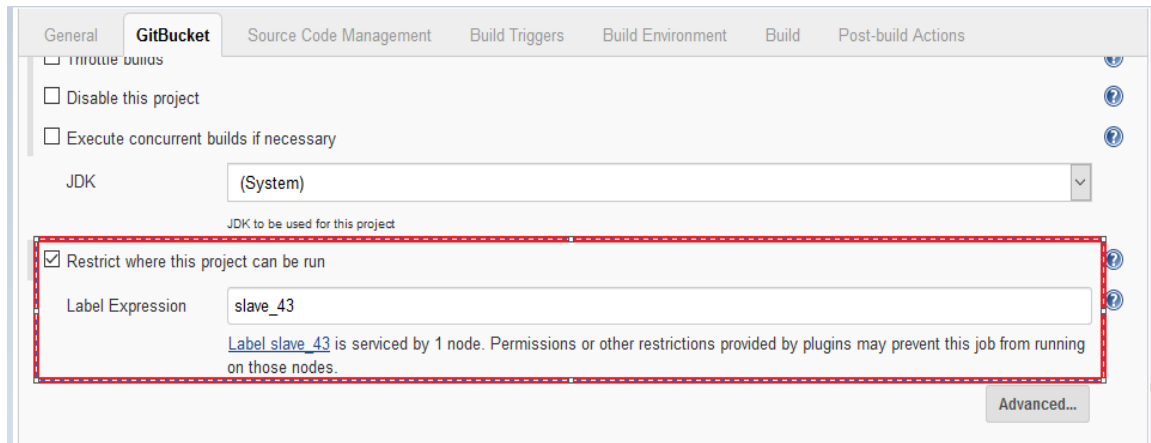
Max # of builds to keep if not empty, only up to this number of build records are kept

Days to keep artifacts if not empty, artifacts from builds older than this number of days will be deleted, but the logs, history, reports, etc for the build will be kept

Max # of builds to keep with artifacts if not empty, only up to this number of builds have their artifacts retained

Thực hiện cấu hình slave để thực hiện quét Sonar:

- Tích vào checkbox ***Restrict where this project can be run*** và nhập tên slave: **slave_43**



Thực hiện cấu hình build:

Di chuyển đến thư mục chứa source trong workspace

Run các lệnh sau:

Npm install

ng build --aot

Lưu ý: version ng 7.2.2 có lỗi memory

⇒ Fix bằng cách run lệnh sau:

node --max-old-space-size=4096 ./node_modules/@angular/cli/bin/ng build --

aot

Zip file (nếu cần) để run deploy:

cd dist/

zip -r ROOT.zip ROOT

Build

Execute shell

Command

```
cd $WORKSPACE/mega-able-v7.2.2-demo
npm install
ng build --aot
cd dist/
zip -r ROOT.zip ROOT
```

See [the list of available environment variables](#)

Advanced...

Add build step ▾

Bước 3: Cấu hình push artifact tới Nexus server

General GitBucket Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Nexus Version

NEXUS3

Protocol

HTTP

Nexus URL

10.60.156.26:8081

Credentials

admin/***** (nexus)

GroupId

\${JOB_NAME}

Version

1.\${BUILD_NUMBER}

Repository

msbuild

Artifacts

Artifact

ArtifactId

\${JOB_NAME}

Type

zip

Classifier

File

\${WORKSPACE}\eshop.zip

Save

Apply

Bước 4: Chạy job và check kết quả

```

+ cd /u01/jenkins/workspace/ETC_IM-UI
+ npm install

> node-sass@4.10.0 install /u01/jenkins/workspace/ETC_IM-UI/node_modules/@angular-devkit/build-angular/node_modules
/node-sass
> node scripts/install.js

Cached binary found at /home/app/.npm/node-sass/4.10.0/linux-x64-64_binding.node

> node-sass@4.11.0 install /u01/jenkins/workspace/ETC_IM-UI/node_modules/node-sass
> node scripts/install.js

Cached binary found at /home/app/.npm/node-sass/4.11.0/linux-x64-64_binding.node

> node-sass@4.10.0 postinstall /u01/jenkins/workspace/ETC_IM-UI/node_modules/@angular-devkit/build-angular/node_modules
/node-sass
> node scripts/build.js

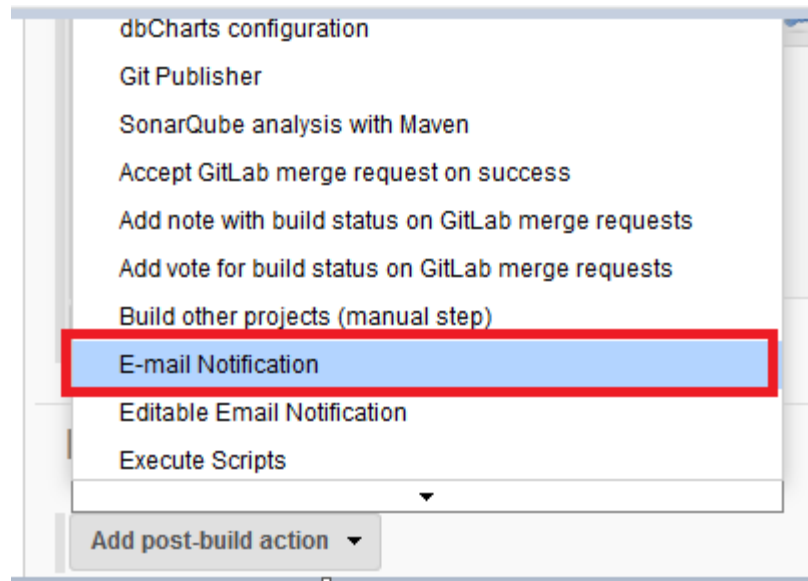
Binary found at /u01/jenkins/workspace/ETC_IM-UI/node_modules/@angular-devkit/build-angular/node_modules/node-
sass/vendor/linux-x64-64/binding.node
Testing binary
Binary is fine

> node-sass@4.11.0 postinstall /u01/jenkins/workspace/ETC_IM-UI/node_modules/node-sass
> node scripts/build.js

```

Bước 5: Cấu hình gửi email thông báo

- Vào Add post-build action → E-mail Notification:



- Cấu hình mail của PM – (người nhận thông báo khi build faild):

Post-build Actions

E-mail Notification

Recipients:

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☒ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build

Add post-build action ▼

NOTE: Hướng dẫn trên chỉ gửi mail khi job chạy failed. Trong trường hợp gửi mail bất kì khi nào job chạy success, failed, unstable thì cấu hình như sau:

- Chọn Post-build Actions → Editable Email Notification:

Post-build Actions

Editable Email Notification

Disable Extended Email Publisher ☐


Project From:

Project Recipient List:

- Click **Advance Settings**:

Attach Build Log:

Content Token Reference:

Click 

Advanced Settings...

- Click **Advance**:

Triggers

Always

Send To: Developers

Recipient List

Add ▼

Advanced...

Add Trigger ▼

- Nhập danh sách người nhận, cách nhau bởi dấu phẩy, có thể chọn trigger gửi khi failed, success, or always

Recipient List: ducnc2@viettel.com.vn, hienptt22@viettel.com.vn

Reply-To List: \$PROJECT_DEFAULT_REPLYTO

Content Type: Both HTML and Plain Text

Subject: \$PROJECT_DEFAULT_SUBJECT

Content: \$PROJECT_DEFAULT_CONTENT

Attachments:

Can use wildcards like 'module/dist/**/*.*.zip'. See the [@includes of Ant files](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log: Attach Build Log

Bước 6: Cấu hình job thực hiện Auto Deploy

Invoke Ansible Playbook

Ansible installation: ansible-2.0

Playbook path: deploy_im_ui.yml

Inventory: ☐ Do not specify Inventory ☐ File or host list ☐ Inline content

Host subset:

Credentials: app (ansible-slave43) Add

Vault Credentials: - none - Add

☐ become ☐ sudo

Advanced...

5. Hướng dẫn cấu hình build với project Android sử dụng gradle build

5.1. Tạo một project tương ứng với dự án

- Tạo một job dạng **Freestyle project**

5.2. Cấu hình job

- Run shell để thực hiện lệnh

Execute Shell

- Thực hiện các lệnh sau:

`gradle wrapper`

`./gradlew build`

- Lưu ý: với một số project đã cấu hình sdk.dir → sửa lại trong file `local.properties` như sau trước khi build các lệnh trên:

`sdk.dir=/home/app/server/android-sdk`

6. HƯỚNG DẪN VIẾT ANSIBLE PLAYBOOK ĐỂ AUTO DEPLOY

6.1. Với project dạng Ant, Maven

- Viết file Ansible Playbook để deploy web app dạng maven trên server tomcat:
File Ansible Playbook có dạng `*.yml`. Nội dung file mô tả các công việc cần thực hiện trên server để deploy một webapp.

Các công việc cần thực hiện bao gồm:

- Copy file build tới server deploy
- Start server tomcat

- hosts: 10.60.156.26

remote_user: nexus

tasks:

- name: copy file war to folder webapp in tomcat

copy:

src: "target/tomcat-test-1.0-SNAPSHOT.war"

dest: "/home/nexus/hddt/tomcat/webapps/app.war"

- name: Go to the folder and execute command start tomcat

command: chdir=/home/nexus/hddt/tomcat/bin nohup ./startup.sh

Trong đó:

- **Hosts:** địa chỉ ip của server deploy
- **remote_user:** user ssh tới server
- **tasks:** các công việc thực hiện trên server

- **name:** tên của một task
- **Một số module được sử dụng để hỗ trợ việc thực hiện tasks:**
 - copy:** module của ansible hỗ trợ việc copy file tới server
 - command:** module hỗ trợ việc thực thi command trong linux

6.2. Với project dạng Msbuild (.NET)

- Đối với việc deploy một webapp trên IIS Server, ansible hỗ trợ nhiều module hỗ trợ để thực hiện các công việc trên windows như: win_copy, win_command, win_domain, win_file, win_get_url, win_unzip, win_iis_website,...

Để deploy web app .net tới server IIS, các công việc cần thực hiện được mô tả trong ansible playbook bao gồm:

- Lấy bản build từ nexus repo
- Uzip bản build tới thư mục wwwroot trong server deploy
- Tạo một website trên IIS tương ứng

- hosts: 10.60.156.203

remote_user: app

tasks:

- name: get file build artifact

win_get_url:

url:

http://10.60.156.26:8081/repository/msbuild/Build_eShop/Build_eShop/1.{{VERSION}}/Build_eShop-1.{{VERSION}}.zip

dest: C:\inetpub\wwwroot\

- name: unzip file build

win_unzip:

src: C:\inetpub\wwwroot\Build_eShop-1.{{VERSION}}.zip

dest: C:\inetpub\wwwroot\eshop

- name: deploy app in IIS Server

win_iis_website:

name: eshop

state: started

port: 9999

ip: 10.60.156.203

application_pool: DefaultAppPool

physical_path: C:\inetpub\wwwroot\eshop\Publish

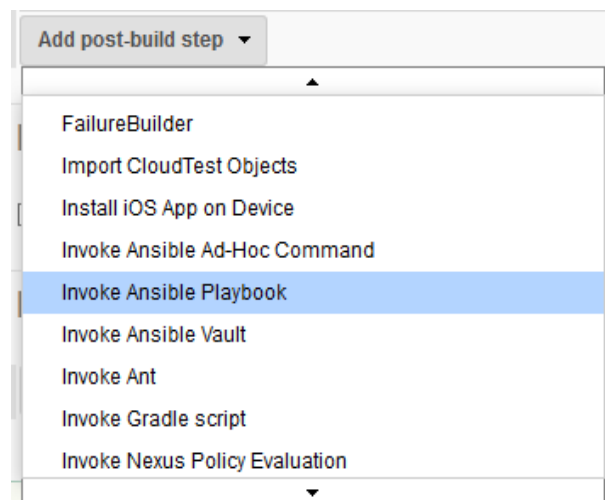
7. HƯỚNG DẪN AUTO DEPLOY TRÊN JENKINS

Lưu ý:

- Với project Msbuild (.NET), cần tạo hai job để thực hiện việc build và deploy do việc build được thực hiện trên jenkins slave window, deploy được kích hoạt bởi server ansible trên jenkins master.
- Với project dạng maven có thể thực hiện trên cùng một job.

7.1. Auto deploy với project maven, ant

- Cấu hình build tương tự mục 1
- Viết ansible playbook tương tự mục 4
- Cấu hình deploy trên jenkins:



- Thực hiện cấu hình như sau:

Invoke Ansible Playbook

Ansible installation: ansible-2.0

Playbook path: deploy_app.yml

Inventory:

- ☒ Do not specify Inventory
- ☐ File or host list
- ☐ Inline content

Host subset:

Credentials: - none - Add

Vault Credentials: - none - Add

☐ become

☐ sudo

Advanced...

Save Apply

- Trong đó:
 - o Ansible installation: phiên bản ansible được cấu hình trong jenkins
 - o Playbook path: cấu hình đường dẫn tới file playbook ansible (file dạng .yaml)

Kết quả chạy job:

```
[test_maven] $ /usr/bin/ansible-playbook deploy_app.yml -f 5

PLAY [10.60.156.26] *****

TASK [Gathering Facts] *****
ok: [10.60.156.26]

TASK [copy file war to server] *****
changed: [10.60.156.26]

TASK [Go to the folder and execute command] *****
changed: [10.60.156.26]

PLAY RECAP *****
10.60.156.26          : ok=3    changed=2    unreachable=0    failed=0

Finished: SUCCESS
```

7.2. *Auto deploy với project Msbuild*

Để deploy một project msbuild cần tạo 2 job riêng, 1 JOB BUILD trên server jenkins slave, và 1 JOB DEPLOY trên server jenkins master.

Có thể tạo một luồng hoàn chỉnh với CI/CD với project Msbuild như sau:

- 1 JOB để quét Sonar
- 1 JOB BUILD project
- 1 JOB deploy project
- Cấu hình quét Sonar (Xem hướng dẫn)

Tại job cấu hình quét Sonar, cấu hình kích hoạt trigger đến JOB BUILD như sau:

Post-build Actions

Trigger parameterized build on other projects

Build Triggers

Projects to build: Build_eShop

Trigger when build is: Stable

Trigger build without parameters: ☐

Predefined parameters

Parameters: ESHOP_WORKSPACE = \$WORKSPACE

Advanced...

Add Parameters

Save Apply Add trigger...

- Trong đó:
 - Project to build: là tên job thực hiện build project đã tạo
 - Parameters: thiết lập parameter sử dụng cho job build. Ví dụ, cấu hình **ESHOP_WORKSPACE = \$WORKSPACE**, job **Build_eshop** sẽ sử dụng workspace của job quét sonar mà không cần phải cấu hình source code.

Cấu hình job build (Xem hướng dẫn mục 3)

- Cấu hình trong job build để sử dụng parameter đã set trong job quét sonar:

☒ This project is parameterized

String Parameter

Name

ESHOP_WORKSPACE

Default Value

Description

[Plain text] Preview

☐ Trim the string

Add Parameter

Cấu hình job để trigger đến JOB thực hiện deploy:

Post-build Actions

Trigger parameterized build on other projects

Build Triggers

Projects to build

deploy_eshop

Trigger when build is

Stable

Trigger build without parameters

☐

Predefined parameters

Parameters

BUILD_VERSION = \$BUILD_NUMBER

Advanced...

Add Parameters

Save

Apply

Add trigger...

Cấu hình deploy job

- Lưu ý: do việc build thực hiện trên server jenkins slave, còn việc deploy thực hiện trên server jenkins master cài ansible server. Vì vậy cấu

hình lấy source có chứa file ansible playbook để deploy project để thực hiện.

- Việc cấu hình lấy source (xem hướng dẫn trong phần build)

Trên jenkins, cấu hình như sau:



Cấu hình lệnh để run ansible, VERSION là biến được sử dụng trong ansible playbook để lấy bản build deploy.

Kết quả chạy job:

```
+ sudo /usr/bin/ansible-playbook test.yml -e VERSION=30

PLAY [10.60.156.203] *****

TASK [Gathering Facts] *****
ok: [10.60.156.203]

TASK [get file build artifact] *****
changed: [10.60.156.203]

TASK [unzip file build] *****
changed: [10.60.156.203]

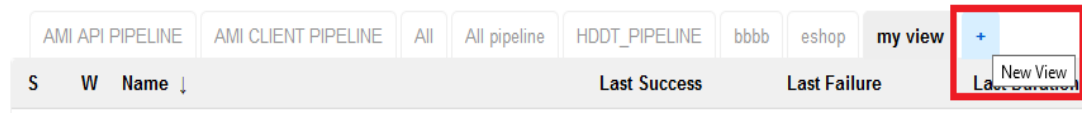
TASK [deploy app in IIS Server] *****
changed: [10.60.156.203]

PLAY RECAP *****
10.60.156.203          : ok=4    changed=3    unreachable=0    failed=0

Finished: SUCCESS
```

Để tạo một pipeline, cho 3 job tạo ở trên thực hiện:

Tạo một pipeline:



Cấu hình:

Name: eshop

Description: [Empty text area]

[Plain text] [Preview](#)

Filter build queue: ☐

Filter build executors: ☐

Build Pipeline View Title: [Empty text field]

Pipeline Flow

Layout: Based on upstream/downstream relationship

This layout mode derives the pipeline structure based on the upstream/downstream trigger relationship between jobs. This is the only out-of-the-box supported layout mode, but is open for extension.

Upstream / downstream config

Select Initial Job: eShop_SONAR

OK Apply

Cấu hình job khởi tạo là job quét sonar.

Kết quả:

